ISSN 2348 – 7968

Detecting Wormhole Attacks Using Cooperative MANET Protocols

S.Keerthika[1] and D.Geetha[2]

[1] Research Scholar, Sree Saraswathi Thyagaraja College Pollachi
Email:keerthisundar89@gmail.com

[2] Assistant Professor Department of MCA, Sree Saraswathi Thyagaraja College Pollachi

***Abstract*— The recent developments in the wireless technology and their Large utilization have made incredible improvements in throughput in the corporate and industrial sectors. However, these recent developments have also introduced new security vulnerabilities. Since the wireless shared medium is totally exposed to outsiders, it is vulnerable to attacks that could target any of the OSI layers in the network stack. For example, jamming of the physical layer, interruption of the medium access control (MAC) layer coordination packets, attacks against the routing infrastructure, targeted attacks on the transport protocol, or even attacks intended to interrupt specific applications. Unfortunately, the effects of applying the security techniques used in wired networks, such as access control and authentication, to wireless and mobile networks have been unsatisfactory due the unique features of such networks. As a result, achieving security goals for mobile ad hoc networks (MANET) has added significant attention in recent years. Many critical applications of MANET, such as emergency rescue operations, military tactical communication, and business operations like mining and oil drilling platforms, require a friendly and cooperative environment.**

***Index Terms*—Wireless technology, MANET, MAC protocol, priority, real-time.**

I. INTRODUCTION

Ad hoc networks are a new standard of wireless communication for mobile hosts. In an ad hoc network, there is no fixed infrastructure such as base stations or mobile switching centers. Mobile nodes that are within each other's radio range communicate directly via wireless links, while those that are far apart rely on other nodes to relay messages as routers.

Mobile Ad Hoc Networks are autonomous and decentralized wireless systems. MANETs consist of mobile nodes that are free in moving in and out in the network. Nodes are the systems or devices i.e., mobile phone, laptop, personal digital assistance, MP3 player and personal computer that are participating in the network and are mobile. These nodes can act as host/router or both at same time. They can form arbitrary topologies depending on their connectivity with each other in the network. These nodes have the ability to configure themselves and because of their self-configuration ability, they can be deployed urgently without the need of any infrastructure. Internet Engineering Task Force (IETF) has MANET working group (WG) that is devoted for developing IP routing protocols. Routing protocols is one of the challenging and interesting research areas for researchers. Many routing protocols have been developed for MANETS.

Wireless ad-hoc networks are composed of autonomous nodes that are self- managed without any infrastructure [1]. In this way, ad-hoc networks have a dynamic topology such that nodes can easily join or leave the network at any time. They have many potential applications, especially, in military and rescue areas such as connecting soldiers on the battlefield or establishing a new network in place of a network which collapsed after a disaster like an earthquake. Ad-hoc networks are suitable for areas where it is not possible to set up a fixed infrastructure. Since the nodes communicate with each other without an infrastructure, they provide the connectivity by forwarding packets over themselves. To support this connectivity, nodes use some routing protocols such as AODV (Ad-hoc On-Demand Distance Vector), DSR (Dynamic Source Routing) and DSDV (Destination-Sequenced Distance-Vector). Besides acting as a host, each node also acts as a router to discover a path and forward packets to the correct node in the network.

II. RELATED WORKS

QNait-Abdesselam et al [2] proposed a wormhole detection method in OLSR protocol which attempts to pinpoint wormhole links before applying the detection algorithm. In addition to OLSR's topology control (TC) message, two new control packets are used: HELLOreq and HELLOrep. The HELLOreq message is used to request an explicit reply from the neighbor. If the HELLOrep from a node is not reached before a predefined timeout interval (Timeout), the originator of the detection process ranks that node as suspicious and stops communication with it until the end of wormhole verification process. To avoid overloading the network with too many HELLOrep, a receiver of HELLOreq delays the replies of multiple requests until it is scheduled to send its normal HELLO message, and piggybacks the replies to this HELLO message.

Van et al. in [3] presents a transmission time based mechanism (TTM) to detect wormhole attacks. TTM detects wormhole during route setup procedure by computing transmission time between every two successive nodes along the established path. Wormhole is identified based on the fact that transmission time between two fake neighbors created by wormhole is considerably higher than that between two real neighbors which are within each other's communication range. In this approach, each node in the established route calculates the RTT between it and the destination and sends this value back to the source. In another approach presented in [4], the authors propose a method called Wormhole Attack Prevention (WAP). It is assumed that each node remains in promiscuous reception mode so that it can always overhear ongoing transmissions. Each node also maintains a Neighbor node table that contains RREQ sequence number, neighbor ID, sending time and receiving time of the RREQ and count. This table is used to monitor the activities of the neighbors. A Wormhole Prevention Timer (WPT) is initiated as soon as a node sends a RREQ.

Maheshwari et al. in [5] proposed a wormhole detection algorithm which looks for forbidden substructure in the connectivity graph that should not be presented in a legal connectivity graph. The authors considered two following communication models for their proposed wormhole detection method:

- Unit disk graph (UDG) model, where each node is modeled as a disk of unit radius.
- General (known or unknown) communication model.

Lee et al. [6] in propose a method which checks whether a node that forwards a packet is a real neighborhood or not. In this approach, each node gathers information of its neighbors within two hops. Each newly joined node broadcasts an announcement which is valid until the next two hops. The requirement of maintaining two types of neighbors, keyed hash and TTL limit the applicability of this method in a distributed system where exists a wide variety of participants

III. Detection of Traditional Wormhole Attacks Using RTT and Topological Comparisons

In presence of traditional wormhole tunnel, the RTT between two fake neighbors is much longer than between two true neighbors. However, longer RTT does not confirm the existence of a wormhole tunnel. This is because other than the wormhole tunnel there are factors (e.g., congestion, intra-nodal processing speed, geographical barrier etc.) That can also contribute to long RTT. In this chapter, a detection method for traditional wormhole attacks is presented. The proposed method detects traditional wormhole attack by using a topological comparison algorithm. [8] Let give the assumption of the Fair Multi-Priority MAC protocol.

3.1 Round Trip Time (RTT) Measurement

In the field of telecommunications, Round Trip Time (RTT) is defined as the time interval between when a packet is sent and when the corresponding acknowledgement is received. One of the most common applications of RTT is finding the best possible route in a communication network. It can range from a few milliseconds (thousandths of a second) under ideal conditions between closely positioned nodes to several seconds under adverse conditions between nodes separated by a large distance. In the context of computer networking, RTT is also known as the ping time which can be determined by using the ping command.

A. RTT Measurement in TCP

One easy way of measuring RTT is to record the time when a packet is sent and calculate the elapsed time when the acknowledgement (ACK) is received. Unfortunately, in TCP there is no way to tell whether a received acknowledgement (ACK) is for an original or retransmitted packet. This is known as “retransmission ambiguity” problem [7]. P. Kern, one of the authors of proposed an algorithm known as “Kern's algorithm” which addresses the problem by ignoring round-trip times of retransmitted packets. In TCP, a sender records how long it takes for a packet to be acknowledged by producing a sequence of RTT samples (s1, s2, s3….). TCP implementations estimate the future RTT of a connection by sampling the behavior of the packets sent over it and averaging those samples into an smoothed round trip time (SRTT). The formula used in SRTT is as follows:

$$SRTT_{i+1} = (\propto \times SRTT_i) + ((1-\propto) \times s_i)$$

B.Our Proposed RTT Measurement Method

Asynchronous Clock

We propose to measure the round trip time (RTT) between a source and its n hop neighbors by broadcasting HELLO packets. The recipients of HELLO packet either rebroadcast it until the nth hop is reached or respond with a unicast HELLOrep. In HELLO packet, the hops_to_leave header is used to indicate the number of hops it should travel. Besides, the broadcast time is recorded by the sender so that RTT can be calculated when a HELLOrep is reached. Each node maintains an exponentially weighted average round trip time (RTTavg) for its n hop neighbors. The RTT and RTTavg are calculated using the following formulae:

$$RTT_i = reciving\ time\ of\ HELLO_{rep} - broadcast\ time\ of\ HELLO$$

$$RTT_{avg(0)} = RTT_0$$

$$RTT_{avg(i)} = \left(\propto \times RTT_{avg(i-1)}\right) + ((1-\propto) \times RTT_i)$$

The difference between the SRTT used in TCP and theRTTavg in our proposed method is in the samples used in the exponentially weighted moving average (EWMA) formula. We use the RTTs between a sender and all n hop neighbors as samples. In the following figures we present a scenario to discuss the calculations of RTT and RTTavg for two hop neighbors (n = 2).

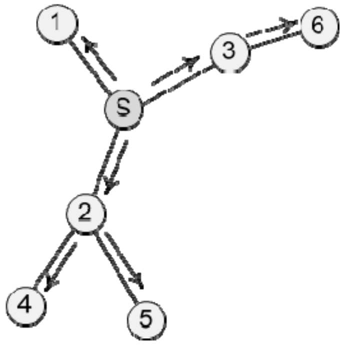


Figure 3.1 HELLO broadcast

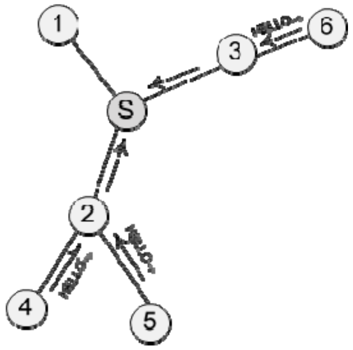
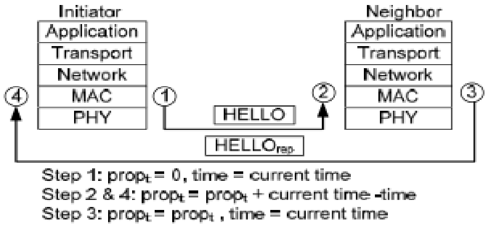


Figure 3.2 HELLOrep unicast

Synchronous Clock

If the clocks are synchronized, RTT can be measured at the MAC layer of the protocol stack. Two additional fields, propt and time, are used in the HELLO packet. The value of propt and time denotes the actual propagation delay and local clock respectively. In this approach, propt and time are updated when a HELLO packet is reached the MAC layer. This approach produces better timing analysis because upper layer delays (specially the delay associated with routing in the network layer) are avoided.



Step 1: $prop_t = 0$, time = current time
Step 2 & 4: $prop_t = prop_t$ + current time -time
Step 3: $prop_t = prop_t$, time = current time

Fig 3.3 Measuring RTT at the MAC layer

In the figure, four major steps of calculating RTT is shown. In step 1, when the HELLO packet reaches the sender's MAC layer, propt is set to 0, time is set to current clock, and then the HELLO packet is broadcasted in the network. In the 2nd step, when the HELLO packet reaches the MAC layer of the receiver, the one way propagation delay from the source to the receiver is calculated and saved in propt, before the HELLO packet is pushed up to the upper layers. Upon receiving a HELLO packet at the application layer, HELLOrep packets are generated. Unlike the HELLO sender, the sender of the HELLOrep attaches the propagation delay calculated in step 2 in propt and then unicasts the HELLOrep back to the source. In this way, upper layer delays can be avoided. Eventually, when the HELLOrep reaches the MAC layer of the initiator, the round trip time is calculated using the same formula used in step 2.

IV. 4. PERFORMANCE EVALUATION

Network Simulator (also popularly called ns-2) is a "discrete event" simulator and is heavily used in ad hoc networking research. It provides necessary support for simulating wired and wireless networks. The ns-2 simulator is coded in two languages: C++ and OTcl. Simulation objects are mirrored in both realms—that means that if one defines a node and some variables associated with a node, the node variables are accessible from code in either language. The intent of this design is to put computationally intensive code in a compiled language (C++), where it can execute fast, while allowing the user to configure the simulator in a more user-friendly scripting language-- in this case, OTcl, or object-oriented Tcl.

A. Generating Topologies in ns-2

Network scenarios have been generated using Tcl scripts. To evaluate the performance of the wormhole detection method, topologies are generated dynamically using random number generator to place nodes within an area ranging from 800m × 800m to 1400m × 1400m. A minimum distance (ranging from 90m to 200m) is maintained between each node in the network. The wormhole attackers are also placed randomly in between two randomly selected (target) nodes in different network segments. The distance between each wormhole attacker is varied on the basis of network area and number of nodes. Another exciting feature of randomly generated scenarios is that the topology changes for every simulation run. Hence, the performance evaluation of the wormhole detection method presented in this study is reliable.

B. Simulating Traditional Wormhole Attack

In the traditional wormhole attack, two colluding nodes tunnel packets from their vicinity and attract as many nodes as possible. In between the colluders there are relay nodes placed to increase the length of the tunnel. The relay nodes are responsible for forwarding packets through the tunnel. It should be noted that both the attackers and the relay nodes remain silent to other participating nodes. This means that they neither participate in any network operations, such as routing, nor respond to neighbor discovery or topological comparison packets. They hide their identities by encapsulating the targeted packets. Two new application layer agents have been designed in ns-2 to simulate the wormhole attack.

C. Performance Metrics

The performance of the wormhole detection method presented in this chapter is measured in regards to the following two metrics:

•Detection rate:
The term “Detection rate” takes into account the number of nodes that are possibly attacked by a wormhole and how many of them are successfully detected. The following formula is used to determine the detection rate:

$$Detection\ Rate = \frac{Total\ no.of\ wormhole\ links\ detected}{Total\ no.of\ wormhole\ links}$$

•Accuracy of alarms: The accuracy of alarm represents the efficiency of the wormhole detection method when it detects possible attacks by using topological comparison. It takes into account the number of links declared as attacked by a wormhole and how many of them are actually affected. The following formula is used to determine the accuracy of alarm:

$$Accuracy\ of\ alarms = \frac{Total\ no.of\ wormhole\ links\ detected}{Total\ no.of\ alarm}$$

The above analysis shows that IFP-MAC protocol supports hard real-time traffic and gives bounded time delay, thus it is a hard real-time MAC protocol.

V. Simulation and Experiment

A. Wormhole Detection in AODV Routing Protocol

The performance of the wormhole detection method is evaluated with AODV routing protocol. The implementation of the detection has been presented. In addition, the performance the method presented in this thesis is compared with the method proposed by Z. Tun et al., T. V. Phuong et al. and the RTT-only phase (not executing the topological comparison phase) of the proposed detection method. The results show that both high detection rate and accuracy of alarms can be achieved with topological comparison based approach. This is because the suspected nodes get a second chance to justify their credibility by exchanging relative positioning with the source.

.

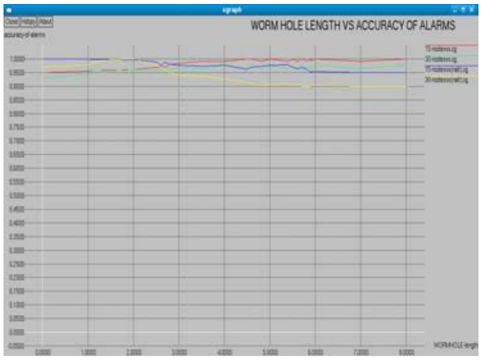


Figure 1. Tunnel length vs. Wormhole detection rate

Fig 1 Shows the detection rate versus tunnel length for different network sizes ranging from 10 nodes to 30 nodes. It can be seen that the detection rate of the topological comparison based wormhole detection approach shows an increasing trend as the length of the wormhole tunnel is increased. This is because that with longer tunnel length the probability of the actually attacked neighbors being included in the Suspected part of the source’s Neighbor List is almost certain due to the long RTT between them. In addition, with larger network sizes more genuine neighbors are likely to be removed from the suspected list and thus increase the detection rate. The detection rate curves are almost identical for larger network sizes because the rate of change in network size is much higher than the rate of change in number of one hop neighbors.

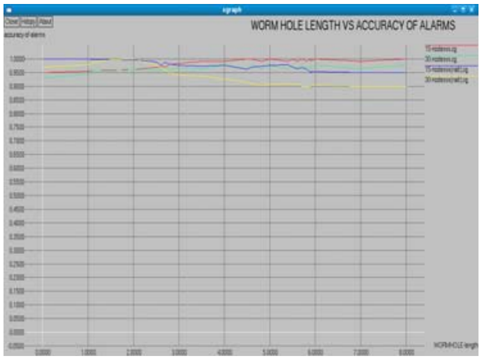


Figure 2Tunnel length vs. Accuracy of alarms

The accuracy of alarm chart is shown as a function of the tunnel length. It can be seen that for longer tunnel length, higher accuracy can be achieved. It is because that, with longer tunnel length, the RTT between a pair of fake neighbors is longer and thus less genuine neighbors to be included in a suspected list. However, we can see a little dip in the accuracy of alarm when the tunnel length is 4-5 and rise again. This may be due to simulation randomness. To demonstrate the effectiveness of the topological comparison, we compare our scheme with RTT-only (i.e., not executing neighbor list comparison). Fig. 3 also shows the accuracy of alarms of RTT-only versus tunnel length for different network sizes. It can be seen that our scheme achieves much higher accuracy of alarms as the topological comparison removes many genuine neighbors from the suspected neighbor list.

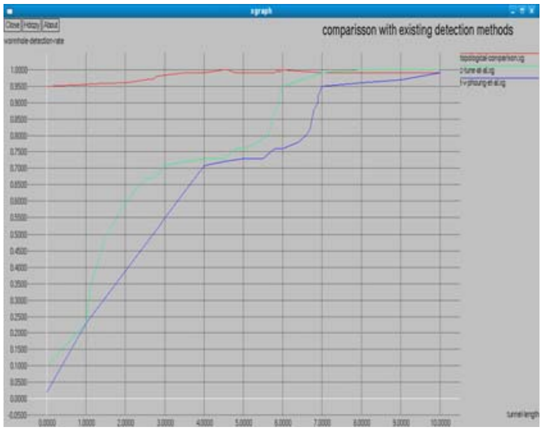


Figure 3 Comparison with existing detection methods (30 nodes)

12 show a performance comparison between the topological comparison based approach and the RTT based methods presented by Z. Tun et al. In [5] and T. V. Phuong et al. In [5]. The topological comparison based approach performs much better when the tunnel length is smaller (e.g., less than 5 hops). As the authors in [4] consider the long RTT between two fake neighbors and the number of neighbors, for smaller tunnels it becomes difficult for this approach to identify the real neighbors from a list of suspected neighbors. However, for larger tunnel lengths their detection rate is identical because the possibility of real neighbors to be included in the suspected list is small.

B. Wormhole Detection in OLSR Routing Protocol

In OLSR routing, each node periodically exchanges link-state messages, such as HELLO and Topology Control (TC). It also uses a multipoint relaying (MPR) strategy, which minimizes the size of the control messages and the number of rebroadcasting nodes. In wormhole attack scenario, an attacker encapsulates the HELLO messages from its vicinity and tunnels them to another attacker. The colluding attacker de-capsulate the tunneled message and then rebroadcast the same HELLO message to its vicinity. For example, in Fig. 3.5, the HELLO messages from nodes s, b and we are tunneled by the wormhole attacker node 1 to the colluder node 3 via the relay node 2. Eventually, nodes d, f and g receive the same HELLO messages. As a result, nodes s, b or e will choose d, f or gas MPR and vice versa. This leads to exchange of some Topology Control (TC) packets through the wormhole tunnel (1-2-3). Since only the MPR nodes are responsible for forwarding TC packets, selecting MPRs that possess flawed network topology may lead to routing disruption and ultimately result in performance degradation of the network as a hole.

The interval between two successive HELLO messages is predefined. We propose that after n number of HELLO transmissions, a node sends Helloed which represents the HELLO for one hop neighbor discovery phase of the wormhole detection method. The subsequent phases of the wormhole detection process are as mentioned in section 3.3. The number n depends on the desired security level. The performance evaluation of the topological comparison based wormhole detection method is presented in Fig 4 and Fig. 5. In Fig. 4, the topological based approach is compared with the method. The graph shows that the detection rate for the topological comparison based method is significantly higher than the other method. In addition, higher accuracy of alarm can also be achieved as shown in Figure 5.

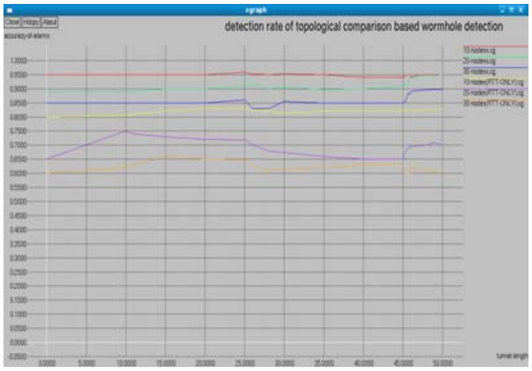


Figure 4 Wormhole length vs. Detection rate

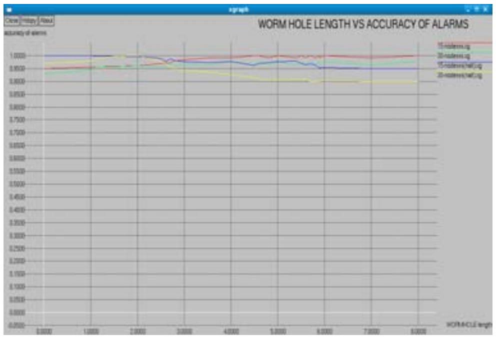
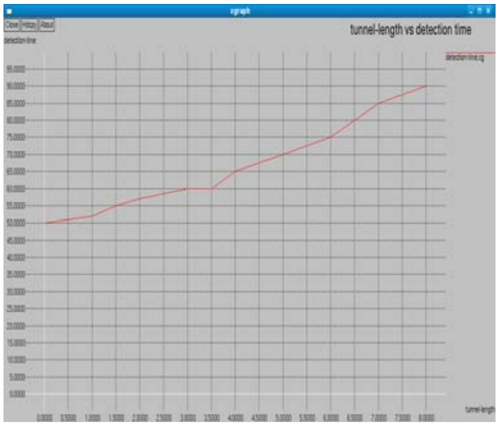


Figure 5 Wormhole length vs. Accuracy of alarm

C. Message Overhead and Detection Time

The topological comparison based method gives the initially suspected nodes a second a chance to verify the reliability. However, this approach introduces some message overhead to the system. In the wormhole detection method proposed in this thesis, we use HELLO, HELLOrep, ENQ, Enwraps messages. Since our proposed wormhole detection method achieves very high detection rate and thus ensures security to the system, the associated message overhead can be tolerated.



Tunnel Length vs. Detection Time

. VI. CONCLUSIONS

A new RTT measurement scheme has been developed to measure RTTs between a node and all its n hop neighbors. The aim of this RTT measurement scheme is to create a suspected n hop neighbor list. Topological comparison based detection method for traditional wormhole detection process creates a suspected one hop list and then runs the topological comparison algorithm. The applicability of this method in AODV and OLSR routing protocols has also been discussed. Besides, the performance of this detection method (in terms of detection rate and accuracy of alarm) is compared with some of the existing methods. The results suggest that the topological comparison based approach performs better than the existing methods. One more topological comparison scheme has been presented to detect wormhole tunnels. Unlike the traditional wormhole detection method, three hop topologies are compared between the originator and its one hop neighbors. The AODV implementation of this method has also been presented. Then, the performance in terms of detection rate and accuracy of alarm has been measured. The results suggest that both high detection rate and accuracy of alarm can be achieved.

REFERENCES

[1] C. Perkins, Ad hoc networking, Addison-Wesley, 2000.

[2] F. Nait-Abdesselam, B. Bensaou, and T. Taleb, Detecting and avoiding wormhole attacks in wireless ad hoc networks. IEEE Communications Magazine, 2008. 46(4): p. 127-133.

[3] T. Phuong, N. Canh, Y. Lee, et. al. Transmission Time-Based mechanism to detect wormhole attacks. In Proceedings of The 2nd IEEE Asia-Pacific Service Computing Conference, 2007.

[4] S. Choi, D. Kim, D. Lee, et. al. WAP: Wormhole attack prevention algorithm in mobile ad hoc networks. In Proceedings of IEEE International Conference on Ubiquitous and Trustworthy Computing, June 2008. p. 343-348.

[5] G. Lee, J. Seo, and D. Kim. An Approach to mitigate wormhole attack in wireless ad hoc networks. In Proceedings of IEEE International Conference on Information Security and Assurance,2008.

[6] R. Maheshwari, J. Gao, and S. Das. Detecting wormhole attacks in wireless networks using connectivity information. In Proceedings of INFOCOM '07, May 2007, p. 107-115.

[7] P. Karn and C. Partridge, Improving round-trip time estimates in reliable transport protocols. ACM SIGCOMM Computer Communication Review, 1995. 25(1): p. 66-74..

[8] X. Su and R. Boppana. Mitigating wormhole attacks using passive monitoring in mobile ad hoc networks. In Proceedings of IEEE GLOBECM '08, December 2008, p. 1-5.