A thesis for a master degree

修士論文

Detection of the Black Hole Attack in a Mobile Ad hoc Network (MANET)

アドホックネットワークにおけるブラックホール攻撃の検知

Graduate School of Frontier Sciences, the University of Tokyo
Department of Frontier Informatics
東京大学大学院 新領域創成科学研究科
基盤情報学専攻

XiaoYang Zhang (47-76318)
張 笑陽

Professor Yasushi Wakahara
指導教員 若原 恭 教授

2009 年 1 月 27 日

Acknowledgements

My deepest gratitude goes first and foremost to Professor Wakahara, my supervisor, for his constant encouragement and guidance. During the past two years, I have learned lots of knowledge from Professor Wakahara. What is more important, he let me know how to be a good researcher. Without his consistent and illuminating instruction, this research could not have reached its present state. Second, I would like to express my heartfelt gratitude to teaching assistant Mr. Sekiya, who have instructed and helped me a lot in the past two years.

I also owe my sincere gratitude to Ms. Li, Mr.Pakornsiri and Mr.Sakamoto who gave me their help and time in helping me work out my problems during the last two years. All the members of the Wakahara lab are thanked for numerous stimulating discussions, help with experimental setup and general advice

Finally, I am also grateful to Ms. Kawasaki and Ms. Yoshizawa, the staff of the Wakahara lab, for their support during the last two years.

.

Content

List of figures

Abstract

Black hole attack is a serious threat in a mobile ad hoc network (MANET). In this attack, a malicious node injects a faked Route Reply message to deceive the source node so that the source node establishes a route to the malicious node and sends all the data packets to the malicious node. Every conventional method to detect such an attack has a defect of rather high rate of misjudgment in the detection. In order to overcome this defect, we propose a new detection method based on checking the sequence number in the Route Reply message by making use of a new message originated by the destination node and also by monitoring the messages relayed by the intermediate nodes in the route. Computer simulation results demonstrate that our method has a feature of much lower false positive and negative rates in detecting any number of malicious nodes than the conventional methods.

1. Introduction

A mobile ad hoc network (MANET) is a self-configuring network that is formed automatically by a collection of mobile nodes without a centralized management. These mobile nodes communicate with each other directly if they are in the same radio communication range. Communication between nodes out of the radio range requires the cooperation of other nodes, which is known as multi-hop communication. Therefore, each node must act as both a host and a router simultaneously. The network topology frequently changes due to the mobility of mobile nodes as they enter, move within, or leave the network.

One of the typical routing protocols for MANET is called Ad hoc On-Demand Distance Vector (AODV) [2], which is defined as a RFC by IETF. In this protocol, if a source node wants to send data packets to a certain destination node, the source node broadcasts a Route Request (RREQ) packet. Every node that receives the RREQ packet checks whether the node is the destination for that packet and if it is the case, the node sends back a Route Reply (RREP) packet. If it is not the case, then the node checks with its routing table to determine if it has a route to the destination. If it does not have such a route, it relays the RREQ packet by broadcasting the packet to its neighbors. If it has a route to the destination, then the node compares the destination sequence number in its routing table with that in the RREQ packet. The number in the RREQ packet was obtained by the source node from the packet that had been transmitted by the destination to the source node before. If the number in the routing table is larger than that in the RREQ packet, the route is fresher and the data packets can be sent through this route. Then this node becomes an intermediate node and sends back a RREP packet to the source node along the route through which it received the RREQ packet. The source node then updates its routing table and starts to send its data packets through this route.

However, this protocol is highly susceptible to routing attacks especially the black hole attack [3] because of the dynamic topology and lack of any infrastructure in the network. There are some types of black hole attack in a MANET. One of the most serious black hole attacks is defined as follows. When an attacker receives RREQ, it returns RREP to the source node with a very large destination sequence number (SN) to make the source node believe that the attacker has the freshest route to the destination. Therefore, the source node will select this forged route to the attacker and discard other legitimate RREPs.

A lot of researchers have proposed their methods to detect and defend against such kind of black hole attack in a MANET. However, to the best of our knowledge, all of the conventional methods have a high rate of misjudgment. Furthermore, few conventional methods cover the case with two or more malicious nodes that cooperate with each other to

perform the black hole attack, which could be quite usual in a MANET.

In this paper, we propose a new method which can be used to cope with the attack and to solve the problems of the conventional methods. The proposed method does not use a public key infrastructure system (PKI) [4] which leads to more complex problems including the key distribution. In our method, when an intermediate node unicasts a RREP message, the node also unicasts a newly defined control message to the destination node to request for the up-to-date SN. Then the destination node unicasts a reply message to inform the source node of the up-to-date SN after receiving the request message sent by the intermediate node. This reply from the destination node enables the source node to verify if the intermediate node has sent a faked RREP message by checking if the SN in the RREP message is larger than the up-to-date SN. Furthermore, this reply can also be used to confirm whether the intermediate node really has a route to the destination node.

This paper is organized as follows. In section 2, we give an introduction of related background knowledge. In section 3, we discuss the related works on the detection of a black hole attack and their weakness to clarify the necessity of a new detection method. We propose a new detection method and its algorithm in section 4. In Section 5, we show and discuss the simulation results to demonstrate the power and the performance of our method. Finally, we give the conclusion in section 6 and the related future works in section 7.

2. Background

2.1 Principles of AODV

In a reactive routing protocol, where a route is established on the basis of demand, a control packet named Route Request (RREQ) message is broadcast by the source node in order to find an optimal route to the destination node. The destination sequence number is an important attribute in Route Request message to determine the freshness of a particular route. Upon receiving the Route Request packet, a node either:

i) replies to the source node with a Route Reply (RREP) packet, if the node is the destination node or an intermediate node with 'fresh enough' route to the destination, or

ii) rebroadcasts the Route Request packet to its neighbors if the node is neither of the above-mentioned destination and intermediate nodes.

An intermediate node is deemed to have a fresh enough route to the destination if the destination sequence number in its routing table entry is greater than or equal to the destination sequence number of the Route Request. Once the source node receives the Route Reply, it establishes a route to the destination. The Route Reply message normally has the value of the Route Request's destination sequence number, which is normally incremented by one by the destination node [1]. Fig. 1 briefly illustrates this process.

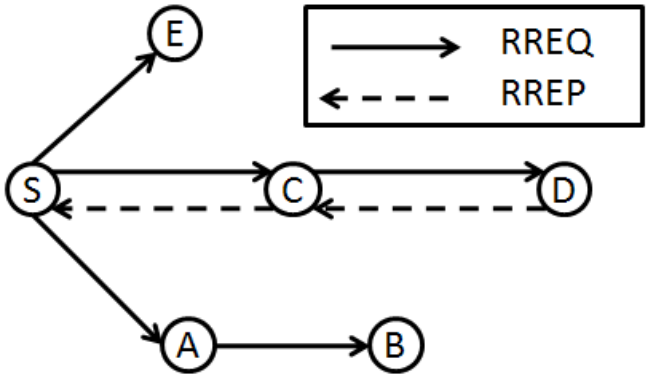


Figure 1An example of AODV operation

In this figure, node S is a source node and it generates a RREQ message and broadcasts it to its neighbors: A, C, and E. This RREQ message contains the last known destination SN. If any of the neighboring nodes has a fresh enough route to the destination node D, it will send back a RREP message to node S. On the contrary, in case where it does not have a fresh enough route to node D, it will rebroadcast the RREQ message, and this activity is repeated until the packet reaches D. When the destination node D receives the RREQ message, it sends back a RREP to S. When node S receives the Route Reply, a route is established. In

case where node S receives multiple RREP messages, it will select a RREP message with the largest destination sequence number value.

2.2 Black Hole Attack

The route discovery process described earlier is susceptible to a black hole attack. The attacker forges its destination sequence number, thus pretending to have the fresh enough route information to the destination. More precisely, upon receiving the broadcasted Route Request message, the attacker creates a RREP message with a spoofed destination sequence number; a relatively large destination sequence number in order to be favored by others. Once the source node receives the reply from the attacker, it routes the data traffic to the attacker. Upon receiving the data packets, the attacker normally drops them and creates a 'black hole', as the attack name implies. Alternatively, this attack can be used as the first step in the man-in-middle attack, where the malicious node may change, delay, delete or manipulate the data packets.

Figure 2 gives an example of such an attack.

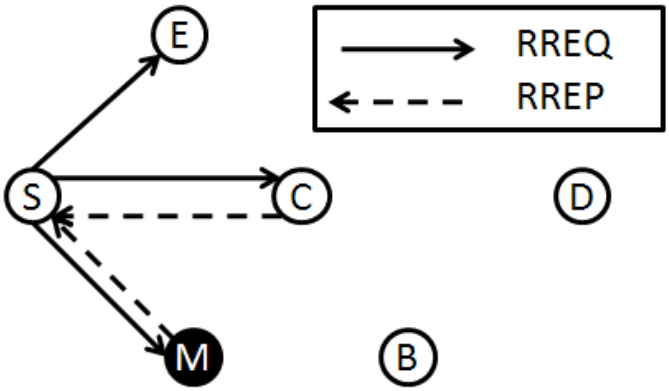


Figure 2 An example of black hole attack

In this figure, source node S generates a RREQ message and broadcasts it to its neighbors: M, C, and E. Suppose node M is a black hole attacker and generates a RREP message with a spoofed SN and sends back this faked message to node S. Meanwhile, node C, which is a normal node, also unicasts a RREP message to the source. After receiving these two RREP messages, node S should make a decision on which route should be selected. The spoofed SN in the RREP message generated by the malicious node M is usually larger than the usual SN, and therefore, the source node S believes that node M has the freshest route to the destination node D and begins to send data packet to node M.

3 Related works

There have been a lot of researches on the techniques to cope with the black hole attack. Some of them, which are considered effective and efficient, are described below and they are followed by the executive summary of their qualitative evaluation results.

A. Method 1: Analytical Model of Route Acquisition Process Approach

Hollick, Schmitt, Seipl and Steinmetz [2] adopted a 2-stage approach. First, they developed a realistic analytical model of the AODV route acquisition process. Second, they extended the work to derive a classification scheme for misbehaving nodes, including nodes of black hole behaviors [3]. Their approach is described as follows:

1) Analytical model of route acquisition process:

This model predicts the probability density function of estimated route lengths, a powerful metric for characterization of the network behavior. The derived probability density function p(d) and the corresponding probability distribution function P(d) are given in equations below. Detailed discussion on the derivation of the equations is given in [2]. The p(d) describes the statistical relations between the distance of two nodes and the corresponding probability of being connected, while P(d) gives the route length distribution in the network. The variable d represents the distance between the source and the destination.

2) Misbehaving nodes effect:

They extend the model to cover the effect of the node misbehavior [3]. That is the deformation of the probability distribution when misbehaving nodes are present. The deformation allows distinction between the normal behavior and the node misbehavior.

$$p(d) = \begin{cases} 2d(d^2 - 4d + \pi) & d \le 1 \\ 8d\sqrt{d^2 - 1} - 2d^3 - 4d) + 4d\left(\frac{1}{\sin(1/d)} - \frac{1}{\cos(1/d)}\right) & 1 < d \le \sqrt{2} \end{cases}$$

$$P(d) = \begin{cases} \frac{1}{2}d^4 - \frac{8}{3}d^3 + \pi & d \le 1 \\ 4\sqrt{d^2 - 1} + \frac{8}{3}\sqrt{(x^2 - 1)} - \frac{1}{2}d^4 + 2d^2 + \frac{1}{3} + 2d^2\left(\frac{1}{\sin(1/d)} - \frac{1}{\cos(1/d)}\right) & 1 < d \le \sqrt{2} \end{cases}$$

B. Method2: Real-Time Intrusion Detection for Ad hoc Networks (RIDAN) System

Stamouli, Argyroudis and Tewari [4] designed a Real-time Intrusion Detection for Ad hoc Networks (RIDAN) system that adopts specification-based detection technique and performs countermeasures to minimise the damage from the attacks. RIDAN details are as follows:

1) Architecture: RIDAN utilizes the timed finite state machines (TFSMs) process, which is an extended finite state machine model with time states and timed constraints on the state transition process. In order to recognize the patterns occurring when an attack is launched, AODV is analyzed in both its normal operation state and its state where an attack is in progress. The timers that control the transition between the states of the TFSMs are derived from theoretical study and practical experimentation.

2) Detection and countermeasure: Based on the TFSMs' design and operation, a node in RIDAN decides if it should either trust another node or must go to an alarm state and take countermeasure against it. The countermeasure action includes isolating the offending node for a finite time period in order to avoid possible false positive. RIDAN implements two different TFSMs to correctly identify the black hole attack.

This TFSM is triggered whenever a node initiates a route discovery process. In state 1, if a Route Reply message does not arrive within a predefined time period (NET_TRAVERSAL_TIME), the TFSM makes timeout (Tout_RESET) and resets the state to its initial state (init_0). Upon receiving the first RREP in state 2, TFSM checks if the included destination sequence number (RREP_dest_seq#) is suspiciously much higher than the sequence number included in Route Request (orig_dest_seq#). If it is suspiciously higher, the TFSM goes directly to the alarm state (Alarm). Otherwise, the TFSM remains in the same state for time t. If the timer expires before receiving another Route Reply, the TFSM resets normally (N_RESET). If within the time limit another Route Reply arrives, the validity of the destination sequence number is checked again in state 3 and similarly a decision is taken whether to move to an alarm state. When an alarm occurs, the source node must not update its routing table with the forged routing information. The next step is to reset (A_RESET) the TFSM to its initial state (init_0).

C. Method3: Neighborhood-based Approach

Sun, Guan, Chen and Pooch [5] developed a neighborhood-based approach to detect as well as respond to the black hole attack. The core of their approach is outlined as follows:

1) Concept: Once the normal route discovery process is finished, the source node sends a

special control packet to request the destination to send its current neighbor set.

2) Neighbor set: The neighbor set of a node is defined as all of the nodes that are within the node's radio transmission range. They claim this metric provides a good "identity" of a node, that is if the two neighbor sets received at the same time are different enough, it can be concluded that they are generated by two different nodes. They verified their claim through the following two experiments:

i) They measured the neighbor set difference of one node at different time instants t and t+1 seconds under different moving speeds and network sizes. The result shows that there is not much change of a node's neighbor set during a route discovery process.

ii) They examined the neighbor set difference of two different nodes at the same time, that is (({A's neighbor set}∪{B's neighbor set}) – ({A's neighbor set}∩{B's neighbor set})). The result shows that the probability that node A's neighbor set is the same as that of node B is very small.

3) Detection: After the source node receives the neighbor set information, it analyses the information by measuring the neighbor set difference. If the difference is larger than the predefined threshold value, the source node knows that the current network has black hole attacks and responds to it accordingly.

4) Response: They proposed a routing recovery protocol with the following two-step approach: i) when a black hole attack is identified, the source node uses a cryptography-based method to authenticate the destination, and ii) once verified, the source node sends a control packet to the destination node to form a correct path by modifying the routing entries of the intermediate nodes between them.

D. Method4: Anomalous Basic Events Approach

Huang and Lee [6] proposed a specification-based approach; to detect violations of the specification directly as well as a statistical-based approach; to detect statistical anomalies by constructing statistical features from the specification in their intrusion detection system. Their anomalous basic events' scheme is defined as follows:

1) Basic concept: A routing process in MANET contains a predetermined sequence of basic events. They utilize Extended Finite State automaton (EFSA), which is similar to a finite-state machine except that transitions and states can carry a finite set of parameters, to specify AODV normal basic events.

2) Basic events: These are also known as basic routing events and are defined as indivisible local segments of a routing process. A Route Discovery process can be decomposed into five series of basic events, starting from the delivery of the initial Route Request by the source node until the reception of the Route Reply message by the source

node and the establishment of a route to the destination. If all of its operations are performed in the specified order, it is considered a normal behavior. In brief, they assume that certain system specification exists to specify normal protocol behavior.

3) Anomalous basic events: An anomalous basic event is a basic event that does not follow the system specification and consists of two components, target and operation. The target can be divided into three categories, which are routing messages, data packets, and routing table (or routing cache).

The taxonomy of the anomalous basic events is represented as a several possible combination of routing targets and operations, as depicted in Table 1. In particular, a black hole attack is considered an attack against integrity with modification on routing messages characteristics in which the attacker changes the sequence number so that some specific route appears more attractive than other valid routes.

E. Method5: Cross-Feature Approach

Huang, Fan, Lee and Yu [7] introduced an anomaly-based detection technique with a data mining capability to automatically construct anomalies detection models using data from trails of network activity. They claim their approach is different from the traditional data mining-based intrusion detection [8, 9] that uses statistical or probabilistic analysis. Instead, their "cross-feature analysis" approach captures the inter-feature correlation patterns in normal traffic such as the relationships between the packets being dropped and route entries being changed. More formally, they study the correlations between one feature and all other features, {f1, f2,..,fi-1,...,fL}→ fi where {f1, f2, ..,fL} is the feature set. Their features include both non-traffic and traffic related features to capture the basic view of the MANET topology and route fabric update frequency. The cross-feature analysis approach comprises the two following phases: Phase 1: Training phase in which classification model Ci is produced from normal data. For all normal vectors (or feature vectors that are related to normal events), they select one feature as the target to classify, which is known as the labeled feature, and then compute a model using all normal vectors to predict the chosen target feature value based on the remaining features. Hence when normal vectors are tested against Ci, it has a higher probability for the true and predicted values of fi to match. In contrast, such probability is significantly lower for abnormal vectors. They name the model as sub-model with respect to fi. They build L sub-models, that is C1 to CL, representing every feature. Here, they adopt some classification algorithms including Ripper [10], C4.5 [11] and Naive Bayes.

Phase 2: Testing phase in which trace logs are analyzed to produce decision threshold. When an event is analyzed, the number of models whose predictions match the true value of the labeled features are calculated. The count is divided by L to produce average match count

and a decision threshold is decided. Hence, any event with average match count below than the decision threshold is considered abnormal. They also propose the use of average probability as an improvement of the average match count approach; i.e. weighted version of the previous approach.

F. Method 6: Dynamic Training Approach

Kurosawa, Nakayama, Kato, Jamalipour and Nemoto [13] also adopted an anomaly-based detection technique but incorporated dynamic training technique. In this approach, the normal state views are updated periodically to adapt to the frequent network changes and 'clustering-based' technique is adopted to identify nodes that deviate from the normal state. They have adopted the following 5-step process:

1) Feature selection: Three features (refer Table 4) are selected to express a normal state of the network. The network state in time slot i is expressed by three-dimensional vector **xi** = (xi1,xi2,xi3).

2) Calculation of mean: The mean vector values of these features are calculated as shown in (1) where D represents a training data set for N time slot.

$$\bar{x}^{D} = \frac{1}{N}\sum_{0}^{N} xi \qquad (1)$$

Hence the initial training data refer to the data collected in the first interval of the network, i.e. ΔT0.

3) Calculation of threshold: For each time slot, they calculate the distance of each input data sample x to the mean vector as shown in (2).

$$d(x) = \|x - \bar{x}^{D}\| \qquad (2)$$

From the learning data set, the distance with the maximum value is extracted as threshold Th.

4) Anomaly detection: When the distance for any input data sample is larger than Th, it is considered deviation from the normal traffic and hence judged as an attack.

5) Dynamic training: By using data collected in initial time 0 ΔT, the calculated mean vector will be used to detect the next period time interval, i.e. ΔT. If the ΔT is judged as normal, the corresponding data set will be used as a learning data set, else, it is treated as data with attack and consequently discarded. This learning process is repeated for every interval ΔT.

Analysis of the above methods

Methods 1, 3, 5 and 6 adopt anomaly-based detection techniques and a node detects any malicious behavior from the pre-established normal profile. In other words, every node in the network provides some related data to other nodes and there must be a threshold that needs to be calculated. These kinds of methods suffer from a high misjudgment rate especially when the definitions of normal behaviors are unclear and are not standardized in wireless ad hoc networks. In contrast, Method 2 and 4 adopt specification-based techniques that monitor the nodes' activities with respect to the pre-defined constraints. This approach has a low misjudgment rate but developing such a specification is time consuming as it needs to be done manually. Moreover, many complex attacks do not violate the specification apparantly and hence cannot be detected using this approach. Therefore, the former anomaly based method is generally preferred, owing to its ability to detect both known and unknown attacks.

Various performance metrics and evaluation results based on these metrics have been presented in the literature, as the authors aim for different objectives. This has made the comparison task more difficult. Therefore, we have selected a few metrics and present them in Table 1 as their executive summary.

Table 1 performance metrics and results

Method	Performance Metrics	Performance Evaluation Results
1	1) probability distribution loss	Analytical model: 8.22% Simulation results: 8.11%
2	1) detection rate 2) delivery ratio	Average detection rate=81.2% Delivery ratio= 55.3%
3	1) detection probability 2) false positive rate	Detection probability= more than 93% False positive rate= less than 1.7%
4	1) Detection rate 2) false positive rate	Detection rate= from 69% to 89% False positive rate=from 24% to 40%
5	Detection probability	Detection probability= from 75% to 95%
6	1) Detection rate 2) False positive rate	Detection rate=more than 80% False positive rate =11%

From the above table, it is apparent that all of the methods have more or less false negatives and false positives. Meanwhile, none of the above methods give a discussion or solution for the cases with more than one malicious node in the network.

Other related methods and their analysis

Some other researchers also have proposed solutions to identify and eliminate black hole nodes [14-16]. In [14], Deng et al. proposed a solution for individual black holes. But they have not considered the cooperative black hole attacks. According to their solution, information about the next hop to the destination should be included in the RREP packet when any intermediate node replies to RREQ. Then the source node sends a further request (FREQ) to next hop of replying node and asks about the replying node and route to the destination. By using this method we can identify trustworthiness of the replying node only if the next hop is trusted. However, this solution cannot prevent from cooperative black hole attacks on MANETs. For example, if the next hop also cooperates with the replying node, the reply for the FREQ will be simply "yes" for both questions. Then the source will trust on next hop and send data through the replying node which is a black hole node. Ramaswamy et al. [15] proposed a solution to defending against the cooperative black hole attacks. But in [15], no simulations or performance evaluations have been done. Ramaswamy et al. [19] studied multiple black hole attacks on mobile ad hoc networks. However, they only considered multiple black holes in which there is no collaboration between these black hole nodes. In this thesis, we evaluate the performance of our proposed scheme in defending against the collaborative black hole attack. In [16], Yin et al. proposed a solution to defending against black hole attacks in wireless sensor networks. The scenario that they considered in sensor networks is quite different from that of MANETs. They consider the static sensor network with manually deployed cluster heads. They did not consider the mobility of nodes. Also they have one sink node and all sensors send all the data to the sink. Each node needs to find out the route only to the sink. Since this scenario is not compatible with MANET, we are not going to discuss it further.

Thus, the disadvantages of all these conventional methods are obvious. First of all, it is difficult to distribute public keys safely in a MANET, which will imply that such a method with PKI is not practical. In the second place, it is not easy to calculate and set an appropriate threshold value and trust level because of the features of the MANET. In addition, most of the above methods can detect only one black hole attacker and do not provide an effective mechanism to cover the situation with more than one attacker in the network. It should be noted that the second and the third disadvantages lead to large false negative rate due to the inability to detect attackers in some cases. To defend against the black hole attack and to overcome the disadvantages listed above, we propose a new detection method based on the destination sequence number without using PKI. Furthermore, our proposed method does not need any threshold nor trust level, and this method can detect more than one attacker at the same time.

4. Proposal of destination sequence number based method

4.1 Assumption

Before describing our proposed method, some assumptions, which are considered realistic, are presented. First of all, the MANET is based on IEEE 802.11 standards. We consider a rather large scale MANET which is deployed in a hostile environment. Nodes are limited in their storage and computational and communication resources. Every node has the same transmission range and non-directional antenna. The nodes are battery-powered, and hence it is crucial to conserve energy to prolong the lifetime of the network. Due to the wireless communication, each node can overhear the message broadcasted by other nodes in the transmission range.

Every node locates randomly and moves randomly, which means the immediate neighboring nodes of any nodes are not known by each other without exchanging any messages. The network is rather dense so that a message in general could be overheard by multiple nodes. We assume that neither the source node nor the destination node is malicious and the adversary who plays black hole attack is an intermediate node.

In addition, we assume there are one or more nodes that perform the black hole attack in the MANET. Moreover, any malicious node has knowledge of all the other malicious nodes' ID and is able to cooperate with these other malicious nodes. The malicious node can change the SN (sequence number) of RREP and contents of legitimate control messages that are newly defined by our proposed method to deceive other nodes.

4.2 Basic detection of single black hole attacker

The black hole attacker is able to inject a RREP message that is faked by changing the SN in the message and to deceive the source node in order to make the source node send its data packets to the attacker. The goal of our method is to protect the network from this attack by detecting the malicious events related to the attack during the route setting up phase. In our method, when an intermediate node unicasts a RREP message, the node also unicasts a newly defined control message to the destination node to request for the up-to-date SN. Then the destination node unicasts a reply message to inform the source node of the up-to-date SN after receiving the request message sent by the intermediate node. This reply from the destination node enables the source node to verify if the intermediate node has sent a faked RREP message by checking if the SN in the RREP message is larger than the up-to-date SN.

Furthermore, this reply can also be used to confirm whether the intermediate node really has a route to the destination node.

The concrete process of detecting the black hole attacker with our method is illustrated in Figure 3 and described as follows:

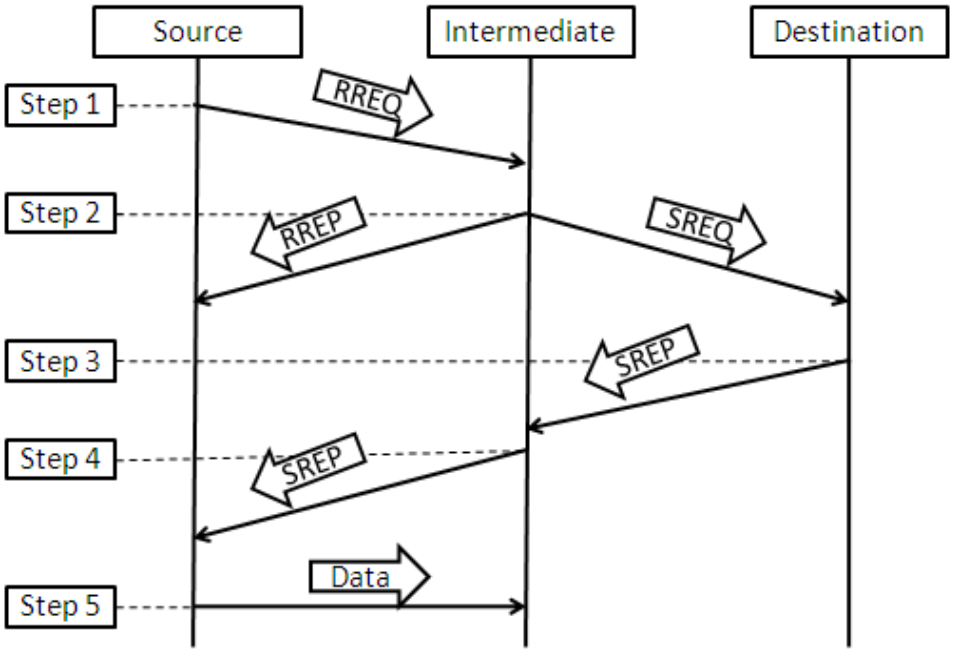


Figure 3 The process of basic detection

Step 1: a source node broadcasts RREQ message to find a route to the destination.

Step 2: an intermediate node which has a route to the destination node receives the RREQ message sent by the source node, the intermediate node generates and returns a RREP message which contains the SN to the source and the intermediate node sends a newly defined SN request (SREQ) message to the destination node at the same time through the route. The source node registers the SN in the RREQ message into its SN table (SNT).

Step 3: the destination node receives SREQ message and sends a SN reply (SREP) message which contains its SN to the source node via the route Destination-Intermediate.

Step 4: the intermediate node relays the SREP message to the source node via the route Intermediate-Source.

Step 5: the source node receives the SREP message, and it compares the SN in RREP message with that in SREP message. If the SN in the RREP message is smaller or equal to the SN in the SREP message, the source node believes there exists no black hole attacker and begins to send data packets.

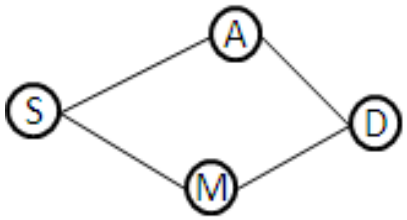


Figure 4 The network topology of Example 1

<Example 1> Figure 4 is an example to clarify the basic idea of the proposed method. There are four nodes in the network which are S, A, M and D.

Assume the source node S broadcasts a RREQ message in order to set up a route to the destination node D. The intermediate node M which receives the RREQ has a route to node D and returns a RREP message with a SN which was obtained by node M when node M received a RREP related to the SN from node D before.

Let the SN in the RREP message received by node S be n_m. Then, node S will register this SN into its SNT as illustrated in Table 2. The second row of the Table 1 is for registering the correct SN obtained from the destination node D.

Table 2 SNT of node S

Node ID	Sequence Number
D	n_d
M	n_m

Meanwhile, node M should also send SREQ message to node D immediately after returning RREP message to node S in order to request node D to send back a message with the correct SN to the source node. When node D receives SREQ message from node M, node D sends a SREP message containing the correct SN to the source node S. Assume that the correct SN of node D is n_d.

When the source node receives the SREP message, it can compare n_m, the received SN in RREP message with n_d.

<End of Example 1>

We analyze the security performance of the above basic process of our method in terms of false positive and negative. The false positive is the case where normal nodes which are judged as malicious and the false negative is the case where all the malicious nodes are not judged as malicious.

False positive: In the process described above, the detection and judgment is executed by the source node in step 5. If the RREP message is generated by a normal node, the SN in the

SREP received by the source node is always larger or equal to the SN in the RREP received by the source node and the source node will not make misjudgment. Therefore, the false positive in this process is equal to 0.

False negative: If there is only one malicious node in the network and if it sends a faked RREP message with a very large SN to the source node, this malicious node will be detected when the source node receives the SREP sent by the destination node in step 5 and thus the false negative is also equal to 0 in this process.

4.3 Detection of single black hole attacker with malicious behavior on SREQ and SREP

In this subsection, we will give a concrete description of how our method works even when the black hole attacker takes malicious behavior during the route setting up phase. As we have already assumed, the malicious attacker node can do any thing on the control messages that are newly defined in our proposed method in order to perform the black hole attack. Since our method does not use the PKI system to e.g. encrypt a message, the malicious node can change any content of the control messages in order to deceive the source node. Therefore, maintaining the consistency of the newly defined messages such as SREQ and SREP is an important issue in our method. It should be noted that any normal node is also able to read the content of the received messages by overhearing them within the transmission range. Thus, we propose a hop-by-hop monitoring mechanism so that not only source node checks and compares the SNs it received but also every overhearing node has a responsibility to monitor all of the messages relayed by the intermediate nodes in the route between the source and the destination nodes.

4.3.1 Detection of SREQ dropping

A malicious node may refuse to generate or relay any newly defined message in order to avoid being detected. However, with our hop-by-hop monitoring mechanism, this kind of malicious behavior can be detected. The concrete monitoring algorithm related to SREQ message is illustrated in Figure 5 and described as follows:

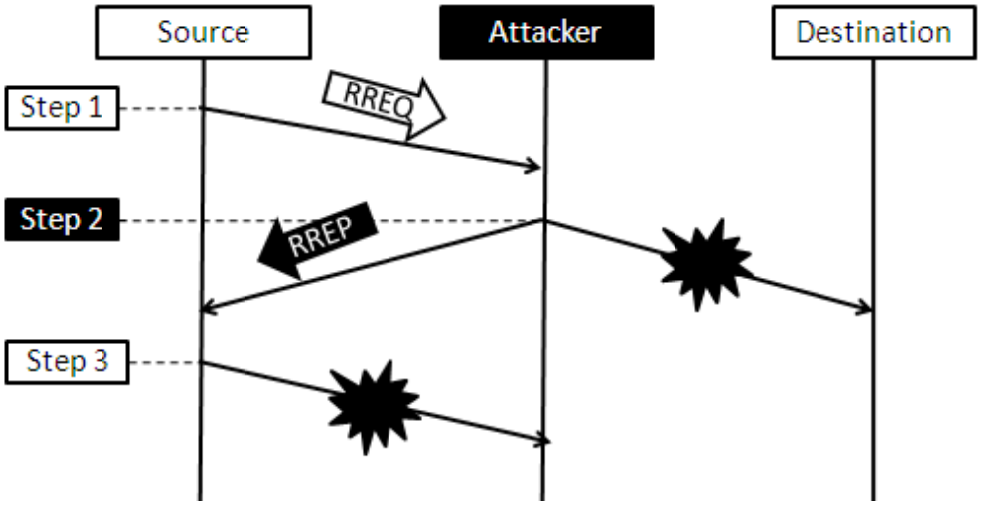


Figure 5 The process of detecting SREQ dropping

SREQ dropping: The intermediate node in Figure 5 is assumed a black hole attacker which receives a RREQ message in step 1 and returns a RREP message with a very large SN. Since SREQ message has a function to confirm whether there really exists a route between the intermediate node and the destination node, this black hole attacker refuses to generate and send SREQ message.

Detection of SREQ dropping: On the overhearing of RREP message originated by this attacker node, all the normal neighbor nodes of the attacker node begin to monitor messages sent from this attacker node. If the attacker node refuses to generate and send SREQ message, none of the other normal neighbors of the attacker node can overhear SREQ message and thus all these neighbor nodes can judge that the intermediate node that originated RREP is an attacker. Then these neighbor nodes broadcast an alarm message to inform the source node of the attacker and at least an alarm message of a neighbor node on the route between the attacker and the source node will reach the source node.

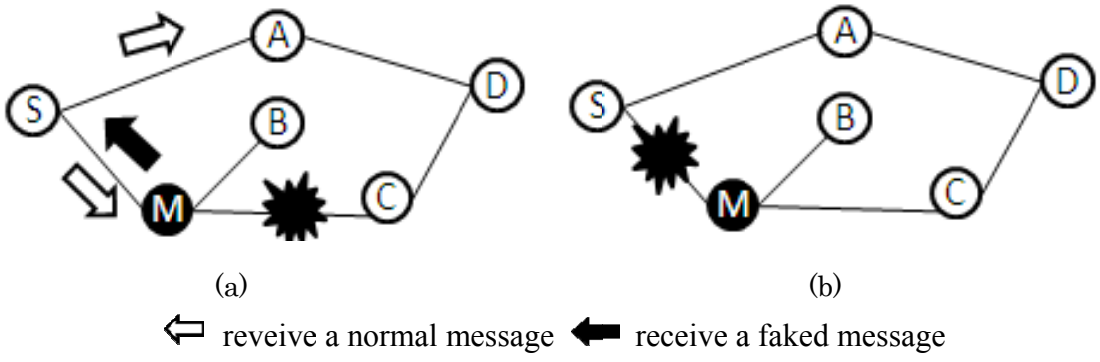


Figure 6 An example of detecting SREQ dropping

<Example 2> In above Figure 6(a), assume that the intermediate node M is a black hole attacker which receives a RREQ and returns a RREP message with a very large SN. Node M has no intention to generate and send SREQ to the destination node. Since node S, a neighbor of node M, will not overhear the SREQ sent by node M even after a while, node S believes that node M is an attacker as shown in Figure 6(b) and selects a route again.

<End of Example 2>

False positive: Unfortunately, the false positive rate is not equal to 0 in this detection of SREQ dropping. We consider the following case illustrated in Figure 7. In Figure 7(a), node A is a normal node which returns a RREP message after receiving a RREQ message broadcasted by the source node S. However, it is possible that when node A generates a SREQ message and begins to unicast it to the destination node D, it has already moved out of the transmission range of node S as shown in Figure 7(b). At this time, node S cannot overhear the SREQ message sent by node A, which will lead to misjudgment that node A refuses to unicast SREQ message to the destination node.

(a) (b)

Figure 7 A false positive example of detecting SREQ dropping

False negative: The monitoring function of a node is triggered when the node receives RREP message from its neighbor node. Therefore, if a malicious node refuses to unicast the SREQ message, this action must be detected by the node that received the RREP. Therefore, the false negative rate in the detection of SREQ dropping is equal to 0.

4.3.2 Detection of change in SN of SREP

Since our method does not use the PKI system to e.g. encrypt messages, so that the malicious node can change the content of a message in order to deceive the source node. The concrete monitoring process for the detection of such malicious behavior related to the relay of SREP is illustrated in Figure 8 and described as follows:

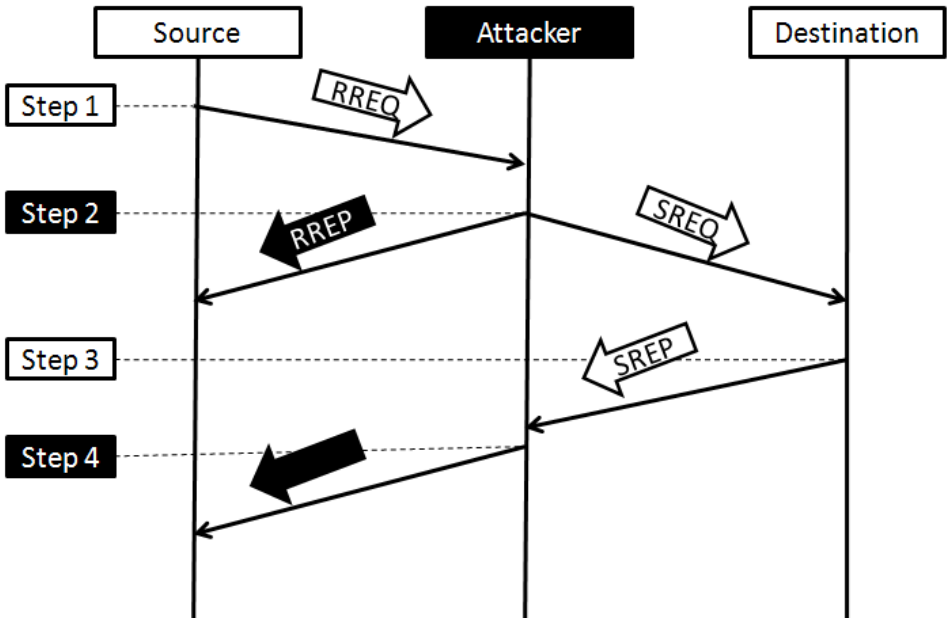


Figure 8 The process of detecting the change in SREP

Change in SREP: In step 2 of Figure 8, the attacker unicasts SREQ message to the destination node and the destination node gives a reply of SREP message with a correct SN to the source node. In order to avoid being detected to be a black hole attacker by the source node, the attacker may change the SN of the SREP message to make the SN equal to or larger than the SN in RREP that was originated by the attacker and send this faked SREP message to the source node. Instead, the attacker may refuse to relay SREP towards the source node. However, this refusal of the relay can be detected by all the neighbor nodes of the attacker in the same manner as the detection of SREQ dropping and thus this refusal will not be discussed hereinafter.

Detection of the change in SREP: When the attacker changes the SN of SREP message and then forwards it to the source node, a neighbor node, if any, of the attacker can detect this change since the neighbor node can overhear this faked message in addition to its original SREP. This neighbor node then broadcasts an alarm message to inform the source node of the attacker.

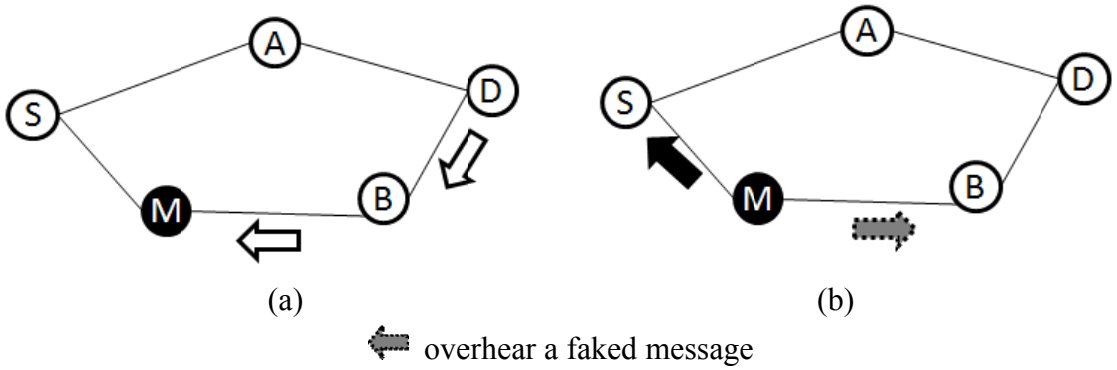


Figure 9 An example of detecting the change in SREP

<Example 3> Suppose that node M sends SREQ message to node D, node D gives a reply of SREP message to the source node and that node B relays this unicast message to node M as shown in Figure 9(a). In Figure 9(b), node M changes the SN of SREP message and then forwards it to the source node. However, node B can detect this change since node B can overhear and read this faked message and node B can judge that node M is a black hole attacker. At this time, node B will broadcast an alarm message to inform node S that node M is malicious.

Therefore, even if an attacker node exists between node M and D and changes the content of the SREP message, the source node can detect the black hole attacker as far as at least one normal neighbor node monitors and checks the contents of the SREPs relayed.

<End of Example 3>

False positive: The false positive rate is 0 in this detection scheme. The reason is that a normal node will never try to change the content of SREP message and so that if the change is detected by a normal node, the node that has made the change is an attacker without doubt.

False negative: When a normal node detects a malicious node by checking the content of SREP, the node will broadcast an alarm message. However, this alarm message may not arrive at the source node in time for various reasons such as no route to the source node. Then, the source node cannot judge that there is a malicious node in the route and begins to send data along this dangerous route, which leads to the false negative. Thus, the false negative rate is not 0.

4.4 Detection of colluding black hole attackers

If there is more than one malicious node in the network, some or all of these malicious nodes

may be located in a route to compose a malicious node chain. Therefore, these malicious nodes can change the content of the SREP message and will not be detected by the normal nodes in the route under setup between the source and destination nodes. In order to detect these colluding malicious nodes, we need some other nodes that are not in the route to play a role of monitoring.

In order to detect colluding black hole attackers, every node in the MANET keeps a SNT. Every node stores the SN contained in the SREP message into its SNT when it overhears the SREP message. Since the SREP message is transmitted along a route, a node which is not in this route may overhear the same SREP message more than one time. The monitoring function of this node is triggered at the first overhearing and this node will compare the SNs of the SREPs it overheard. If the SNs are not equal, this node judges that the SN of SREP is changed and it will broadcast an alarm message to inform the source node of the judgment results.

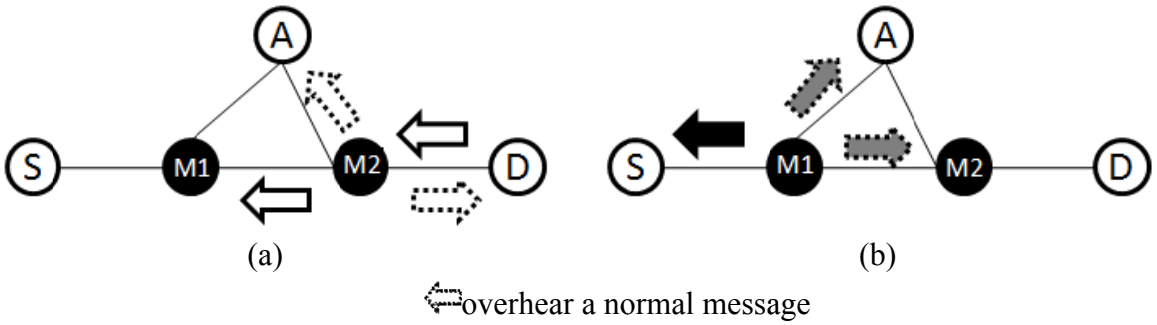


Figure 10 An example of detecting colluding attackers

<Example 4> As shown in Figure 10(a), node A first receives SREP with a correct SN relayed by node M2. However, after a moment, node A overhears another SREP message relayed by node M1 with a different SN as shown in Figure 10(b). Thus, node A can detect the change in SN of SREP even when there are two attackers in the designated route.

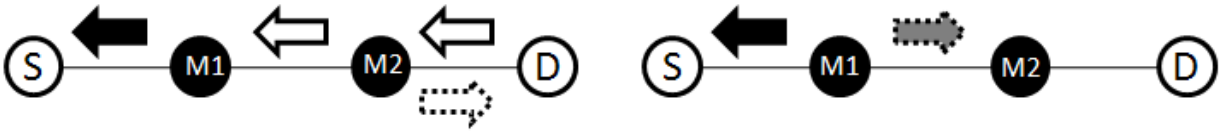
<End of Example 4>

From the above example, we can derive the necessary condition of detecting more than one malicious node in a selected route. A route between two consecutive malicious nodes is called dangerous and a route between a malicious node and its normal neighbor node or a route between two consecutive normal nodes is called safe. Then, the necessary condition for the detection is as follows: if a message is relayed by a malicious node to another malicious node along a route which does not contain any dangerous routes, both of the malicious nodes can be detected. In Figure 10, if the SREP message can be transmitted along the route M2-A-M1, which is safe, then, both malicious nodes in this route can be detected when they change the

content of the SREP.

False positive: The false positive rate is 0 in this detection scheme. The reason is the same as that for the detection of the change in SN of RREP.

False negative: Unfortunately, there are some false negatives, where malicious nodes in the selected route between the source and the destination nodes cannot be detected. Figure 11 shows an example of such false negatives.



(a) (b)

Figure 11 A false negative example of detecting colluding attackers

<Example 5> In Figure 11, node M2 receives the SREP from node D and unicasts this message correctly to node M1 as illustrated in Figure 11(a). Then, node M1 can change the content of SREP relayed by node M2. However node S and node D cannot detect this change as shown in Figure 11(b).

<End of Example 5>

4.5 Reselection of route and transmission of data packet

In this subsection, we will give a concrete description about the issues of reselection of route and the transmission of data packets which are not discussed in most of the previous works related to the detection of black hole attack.

Proposed method 1:

In this method, if there is no malicious node detected in the selected route, the source node begins to send data packets after receiving the SREP message. If there is a malicious node in the route and this malicious node is detected by a monitor node, the monitor node will broadcast an alarm message to inform the source node of the malicious node. After receiving the alarm message, the source node rebroadcasts the RREQ and executes the detection process of black hole attack described in section 4.2

Proposed method 2:

In this method, the source node begins to send data packets as soon as it receives the RREP

message. Every intermediate node should copy and store the received data packets and relays these data packets to the next hop along the route. After checking the SREP message sent by the destination node and confirming no malicious node in the route, the intermediate node stops copying and storing the data packets. If there is a malicious node in the route and a neighbor of this malicious node detects this malicious node by the monitor function described in section 4.2, the neighbor node can broadcast a RREQ message on behalf of the source in order to find a route to the destination node without the malicious node.

The concrete process of data transmission with proposed method 2 is illustrated in Figure 12 and described as follows:

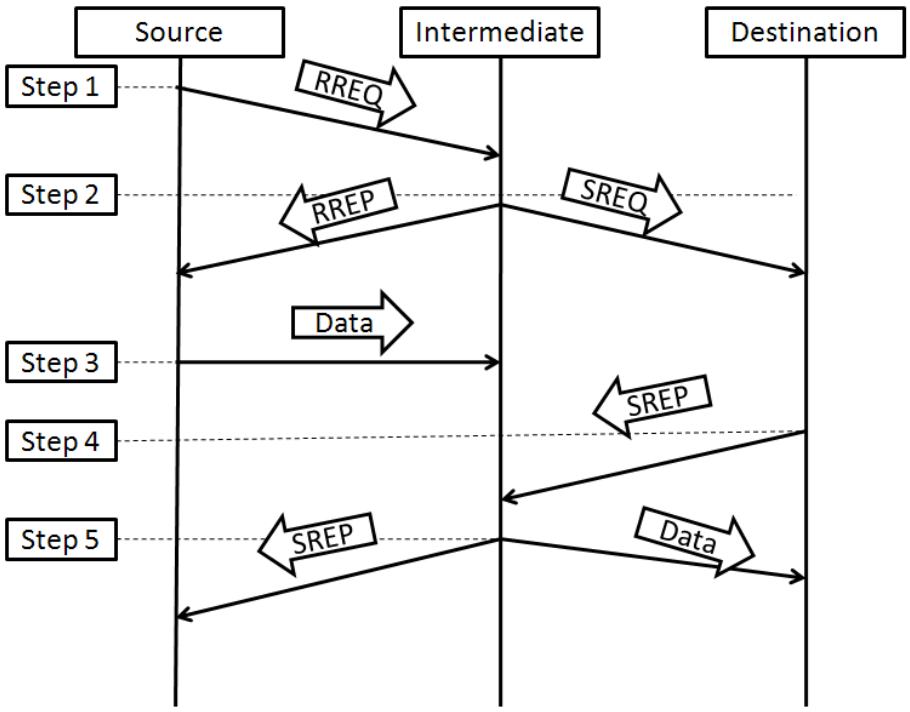


Figure 12 The process of data transmission in method 2

Step 1: a source node broadcasts RREQ message to find a route to the destination.

Step 2: an intermediate node which has a route to the destination node receives the RREQ message sent by the source node, the intermediate node generates and returns a RREP message to the source and the intermediate node sends the SREQ message to the destination node at the same time through the route.

Step 3: the source node receives RREP message and begins to send data packets to the intermediate node. The intermediate node forwards the data packets to the destination node as soon as it receives.

Step 4: the destination node receives SREQ message and sends back a SREP message which contains its SN to the source node via the route Destination-Intermediate.

Step 5: the intermediate node relays the SREP message to the source node via the route Intermediate-Source.

4.6 The overall algorithm of our proposed method

From the processes and examples of the proposed method in 4.2 through 4.4, a node in general can play four roles which are the source, destination, intermediate and monitor. In practice, a node can play a role of monitor with one of the other three roles together at the same time. In other words, every node in our method can play a role of monitor. The action algorithms for the four roles are as follows:

Algorithm 1: role of monitor

1. **when** receive SREQ **do**
2. unicast SREQ to the destination node
3. monitor-1 () {
4. **if** do not overhear SREQ
5. broadcast alarm message
6. }
7. **when** receive SREP **do**
8. unicast SREP to the source node
9. store (ID, SN) of this SREP in SNT
10. monitor-2 () {
11. **when** overhear SREP do
12. compare the SNs
13. **if** SN is changed
14. broadcast alarm message
15. }
16. **when** receive RREP **do**
17. unicast the RREP to the source node
18. if the neighbor node generates the RREP
19. monitor-1 ()
20. **when** overhear SREP **do**
21. store (ID, SN) of this RREP in SNT
22. monitor-2 ()

The algorithm 1 describes the actions of the monitor role. In our method, every node can play a role of monitor. The execution of monitor function depends on the route control message received or overheard. When a node receives the SREQ message, this node unicasts the message and begins to execute the monitor-1. The function of monitor-1 is to monitor whether the next-hop refuses to unicast the SREQ message or not. If this node receives a SREP message, the node will store the SN contained in this SREP message as shown in lines 8 and 9. This node may receive another SREP message some time later and at this time this node executes monitor-2 function to check whether the SN contained in the later SREP has been changed or not from line 12 to line 14. An alarm message will be broadcasted by this intermediate node if the SN has been changed. From line 16 to line 19, when the monitoring node receives a RREP message, the monitoring node will execute the monitor-1 to check if this RREP is generated by its neighbor node and if the neighbor will send a SREQ message or not. If the monitoring node overhears a SREP message for the first time, this monitoring node will store the SN of the SREP message into its SNT as shown from line 20 to line 21. Since this monitoring node may overhear the same SREP message again, this monitoring node can check whether the content SN of SREP message has been changed or not by a node as shown in line 22.

Algorithm 2: role of source

1. generate RREQ
2. broadcast RREQ
3. **when** receive RREP **do** {
4. **if** it is the destination that originated the RREP then
5. send data
6. **if** it is the non-destination neighbor node that generated the RREP
7. store (ID, SN) into SNT
8. monitor-1()
9. check (){
10. **if** receive SREP again later {
11. compare SNs
12. **if** (SN in SREP)<(SN in RREP)
13. discard the RREP
14. **else** send data
15. }
16. **else** discard RREP
17. }
18. **else** store (ID, SN) into SNT
19. check()
20. }

The actions of the source role are shown in algorithm 2. First, the source node generates RREQ and broadcasts this message. After some time, if the source node receives a RREP sent by the destination node, it begins to send data to this destination. However, if the source node receives a RREP generated by an intermediate node, this source node first stores the ID and SN contained in this RREP into the SNT of the source node and waits for the SREP sent by the destination. After the arrival of the SREP at the source node, the source node compares the SN in this SREP with the SN stored in the SNT. By the definition of SN maintenance, the SN in the SREP should be the largest, and thus if the SN in the RREP is larger than that in the SREP, RREP is taken as generated by a black hole attacker.

Algorithm 3: role of intermediate

1. **when** receive RREQ do
2. **if** there is a route to the destination node
3. unicast RREP to the source node
4. unicast SREQ to the destination node
5. **else** rebroadcast RREQ
6. **when** receive RREP do
7. unicast RREP to the source node
8. **when** receive SREQ do
9. unicast SREQ to the destination node
10. monitor-1 ()
11. **when** receive SREP do
12. store (ID, SN) of this SREP in SNT
13. monitor-2()

The algorithm 3 describes the actions of the intermediate role. From lines 1 to 3 and from lines 5 to 7, the intermediate node executes the actions defined as a part of AODV. From line 8, the intermediate node begins to execute monitoring of our proposed method. In this algorithm, the node keeps a SNT. If this intermediate node receives a RREP message, it store the SN into its SNT and unicasts this RREP to the next hop as shown from lines 6 and 7. If this node receives SREQ message and SREP message, this node will play the monitor role in the same manner as described in Algorithm 1.

Algorithm 4: role of destination

1. **when** receive RREQ do
2. send RREP
3. **when** receive SREQ do
4. send SREP contain SN
5. monitor-1()

The algorithm 4 describes the actions of the destination role when it receives RREP and SREP. The node executes the monitor-1 in order to check whether the next-hop changes the contents of the SREP or not in line 5.

5. Evaluation of the Proposed Method

5.1 ns-2 simulator

Ns-2[22] is a discrete event simulator targeted at networking research. It provides substantial support for the simulation of UDP/TCP, routing, and multicast protocols over wired and wireless (local and satellite) networks. We use ns-2 simulator in our simulation.

We first gives a brief overview of the wireless and mobile networking support in ns-2, originally developed as extensions to ns-2 by the CMU Monarch group in the late 1990's [7] and later integrated in the mainline code. The primary purpose of this extension was to enable simulation of wireless networks, in particular multi-hop ad-hoc networks, to compare how different ad-hoc routing protocols performed under various conditions and how they reacted to topology changes. Since then, elements have continuously been added or modified to get a yet more accurate modeling. For instance, currently, a new IEEE 802.11 MAC layer with support for multi-rate options for IEEE 802.11a/b, 802.11e functions to provide service differentiation, as well as new radio propagation models, are being defined and implemented within the Planet project at INRIA [23].

In the ns-2, Wireless components follow the ISO network stack. The physical layer includes radio propagation models, radio interfaces with adjustable parameters such as transmission power and receiver sensitivity and antennas models. The link layer includes medium access control (MAC) protocols needed in such environment, in particular an implementation of the IEEE 802.11 MAC protocol Distributed Coordination Function (DCF), and an implementation of the address resolution protocol (ARP). Wireless routing protocols, e.g. AODV or DSR (Dynamic Source Routing), are implemented as agents according to the ns- 2 terminology. All these entities are linked up together inside the MobileNode, the core object for wireless simulation in ns-2. It is a split object (Tcl/C++) derived from the base Node object. The basic structure of a MobileNode adapted from ns-2 documentation [20] is presented in Figure 13. Optional tracing objects are omitted for clarity. During a simulation study, a MobileNode has a position and may move on topography. According to the documentation, it has one or more wireless interfaces, each of which is attached to a wireless channel. When simulating wireless nodes with limited power, e.g. when simulating wireless sensor networks, a Mobile- Node has also an energy level which decreases with time and the number of packets it handles.

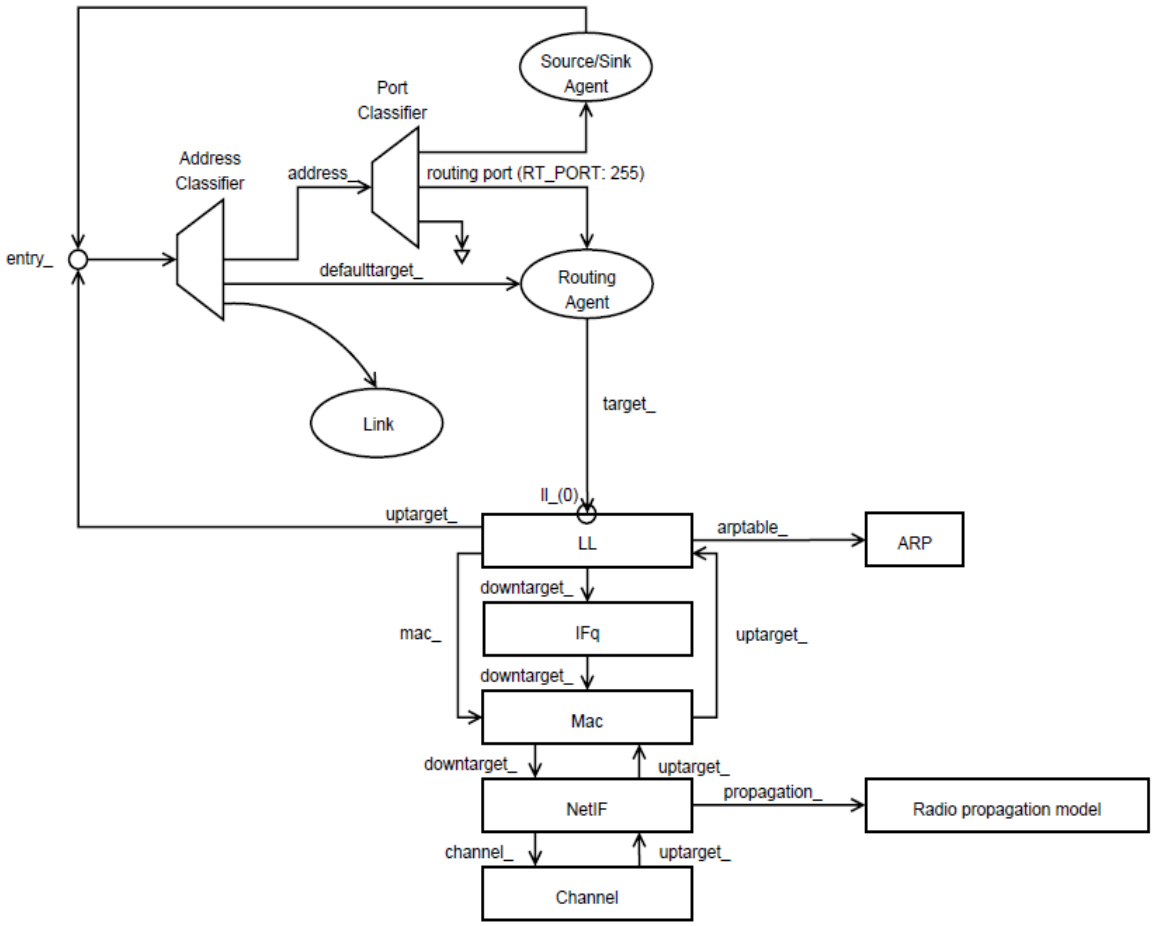


Figure 13 Schematic representation of a MobileNode (CMU Monarch implementation)

5.2 Simulation settings

In this section, we present a set of simulation experiments to evaluate this proposed detection method by comparing it with the original AODV [1] and the solution proposed by [13] (SAODV for short). Compared with our method that gets the SN from the destination node as the threshold, this previous method [13] uses a data learning scheme to make every node have knowledge of the current value of SN by the exchange of routing control messages such as RREQ and RREP. Every node calculates and maintains the average value of SNs in the routing control messages and the value is taken as the current SN. Therefore, if a node receives a RREP message with a SN much larger than a predetermined threshold plus the current SN value, this node will believe that the RREP message is generated by a malicious node. Thus, the detection in this previous method is entirely different from our method.

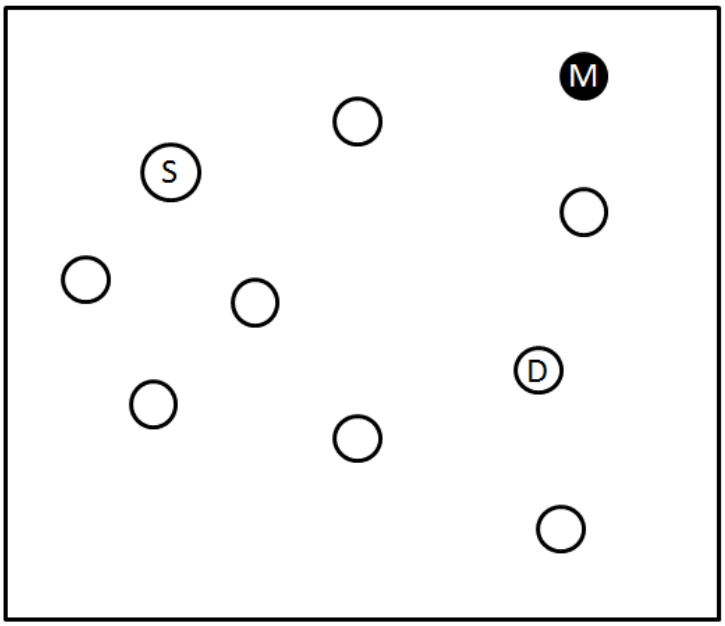


Figure 14 An example of simulation network topology

First of all, we use the Random Number Generator (RNG) supplied by the ns-2 to generate the coordinates of every node in a square as shown in Figure 14. The random waypoint model is used to model mobility. Each node starts its journey from a random location to a random destination. When the destination is reached, another random destination is targeted without pause. We vary the mobile speed according to the scenarios. We distribute the sequence number to every node in the simulation as their identifiers and we always set node 0 as the source node and node 1 as the destination node, respectively. We simulate 200s for each run and the source node begins to broadcast the RREQ message at the time of 150s. We use Constant Bit Rate (CBR) session in the mobile ad hoc network. Each of these CBR applications uses 512-byte data packets at the rate of 1Mb/s. We conduct 1000 independent simulation runs for each scenario with 1-1000 seed values to obtain the average measures for the performance metrics. The following performance metrics are taken in this evaluation.

Unless explicitly said, all simulation scenarios are configured according to Table 3.

Table 3 Simulation Scenario

Simulator	Ns-2 (Ver.2.27)
Number of nodes	30
Number of malicious nodes	0, 1
Area size	1000m×1000m
Transmission range	250m
Speed of nodes	5m/s—30m/s
Node mobility model	Random waypoint
Pause time	0

5.3 End to end time delay

Time delay of the data packets, denoted by T, is calculated as T=T1-Trq, where T1 is the point time when the first data packet is started to receive by the destination node and Trq is the point time when the source node starts to broadcast a RREQ.

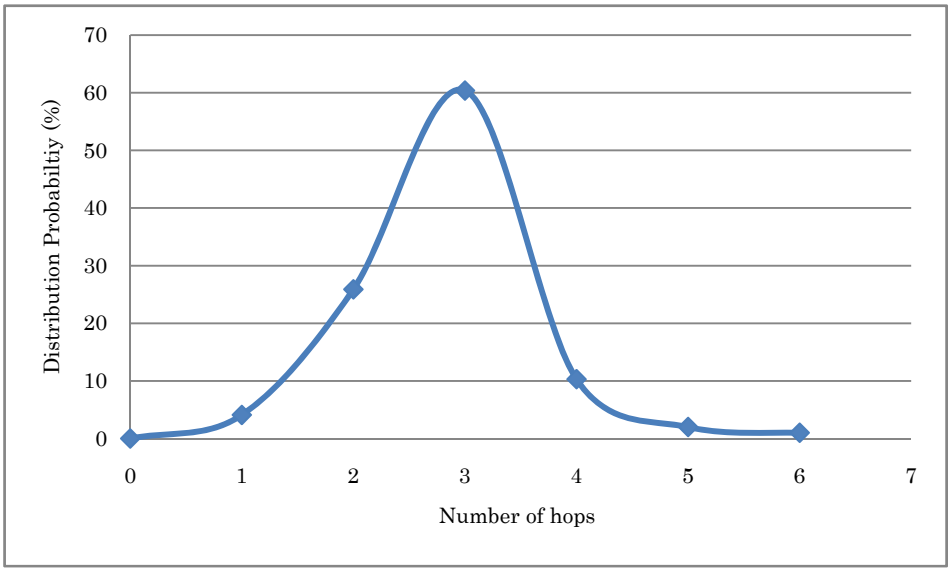


Figure 15 Probability Distribution of Number of hops

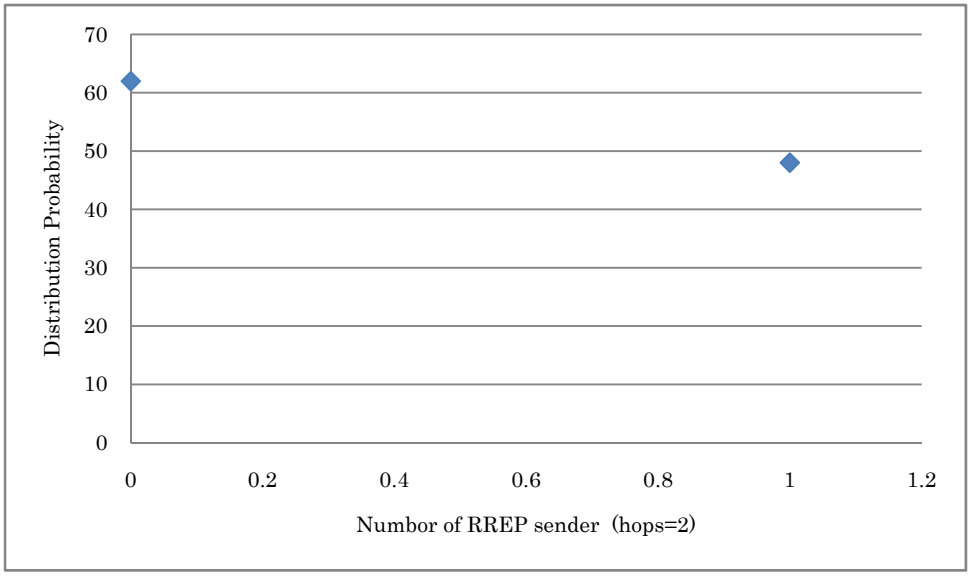


Figure 16 Probability Distribution of Number of RREPs senders (the number of hops=2)

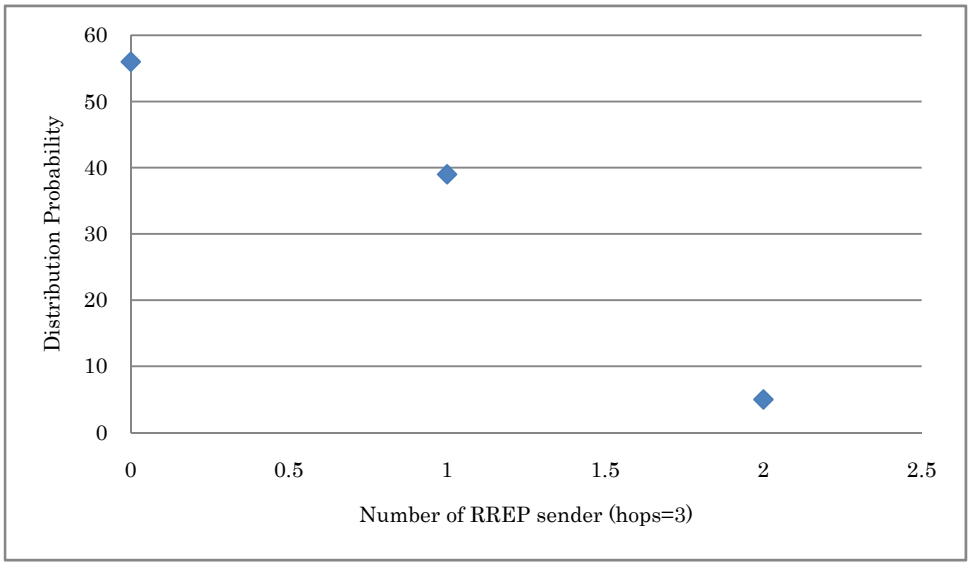


Figure 17 Probability Distribution of Number of RREP senders (the number of hops=3)

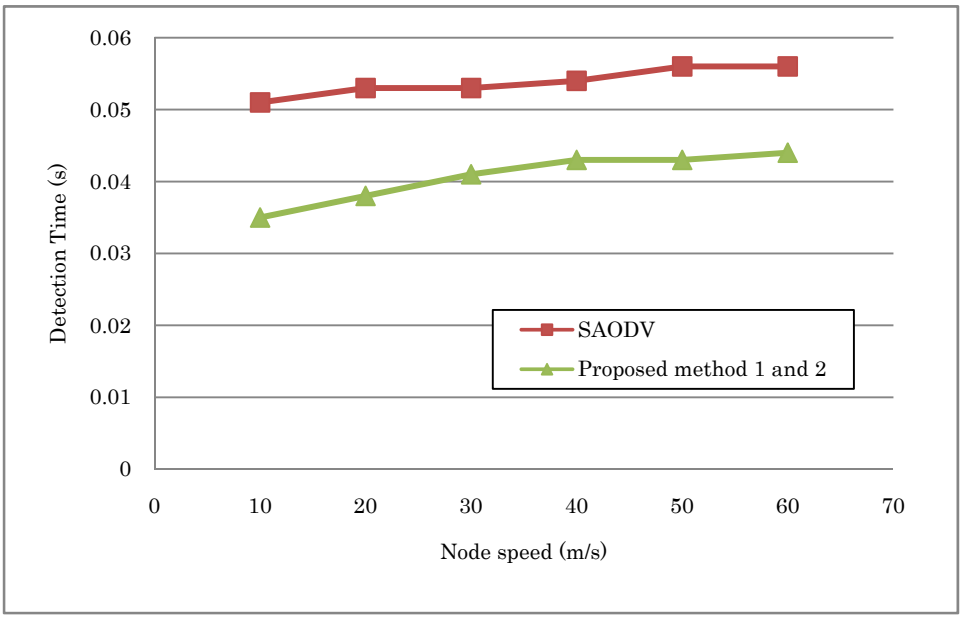


Figure 18 Detection time vs node speed (SREQ dropping)

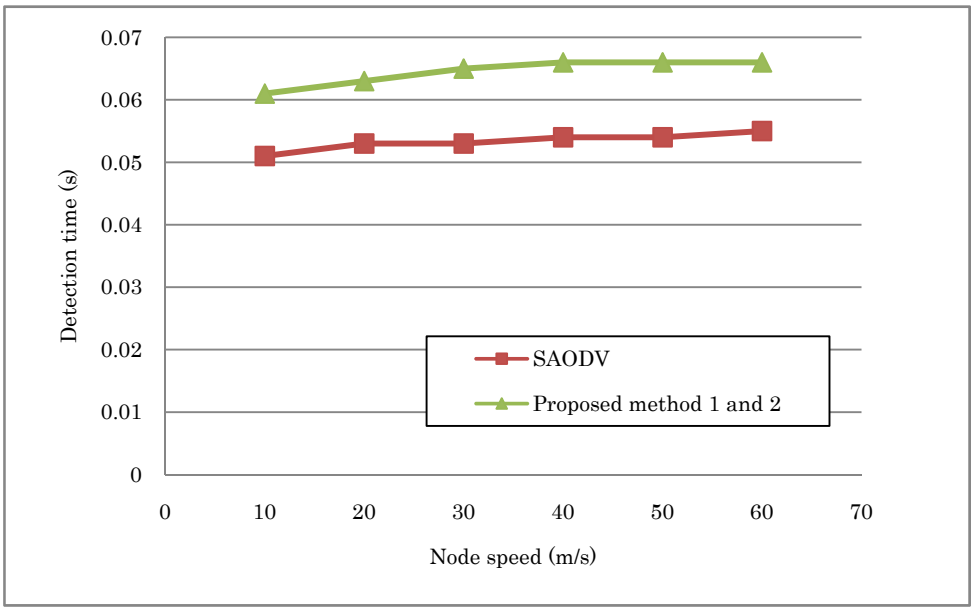


Figure 19 Detection time vs node speed (SN spoofing)

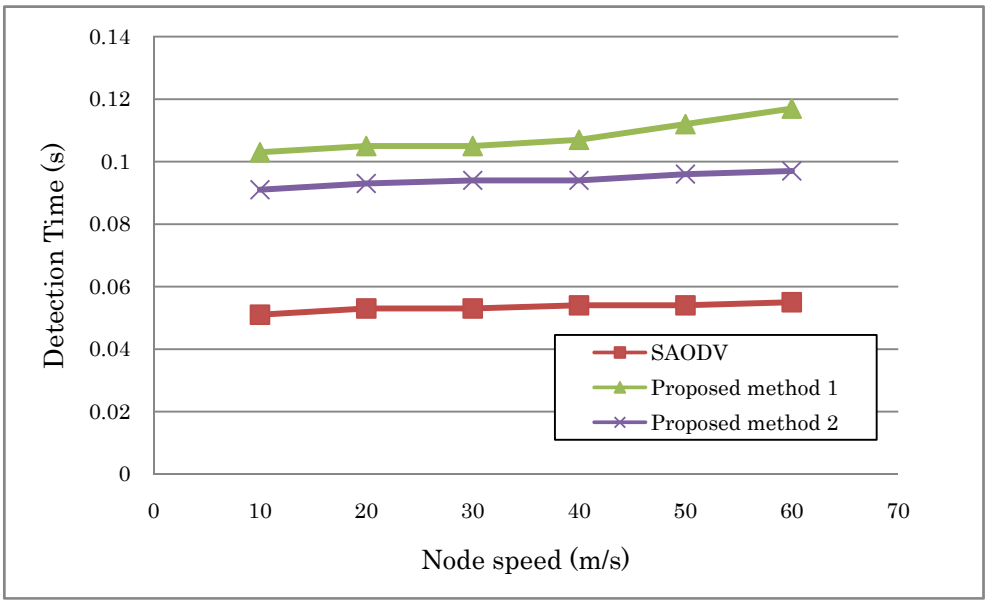


Figure 20 Detection time vs node speed (SREP changing)

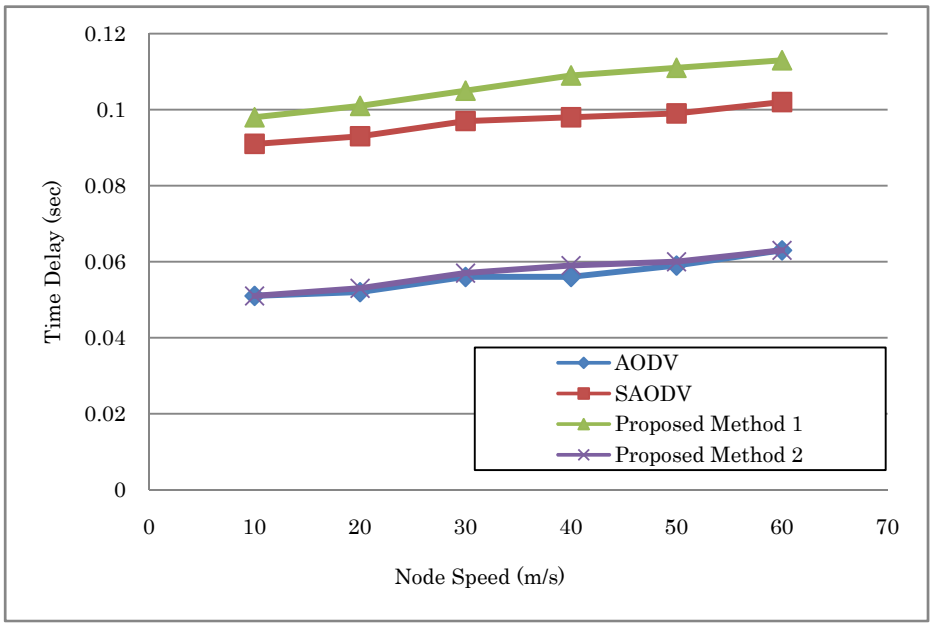


Figure 21 Time delay vs node speed (no malicious node)

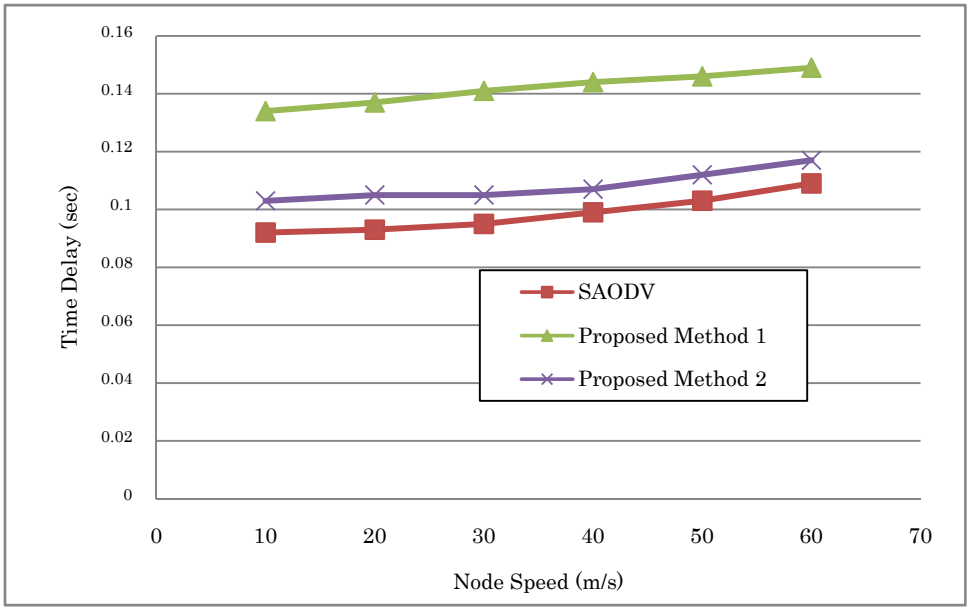


Figure 22 Time delay vs node speed (one malicious node)

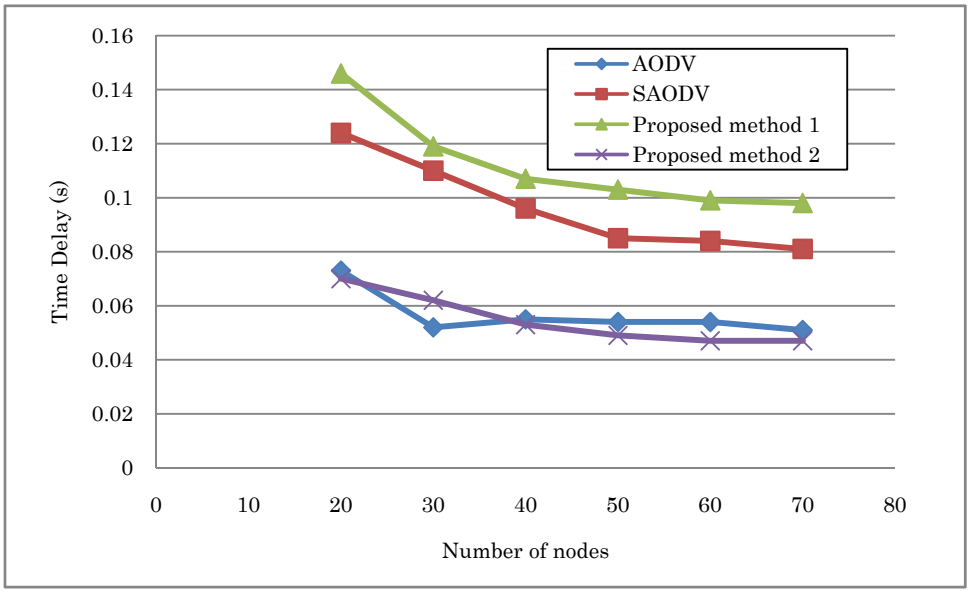


Figure 23 Time delay vs number of nodes (no malicious node)

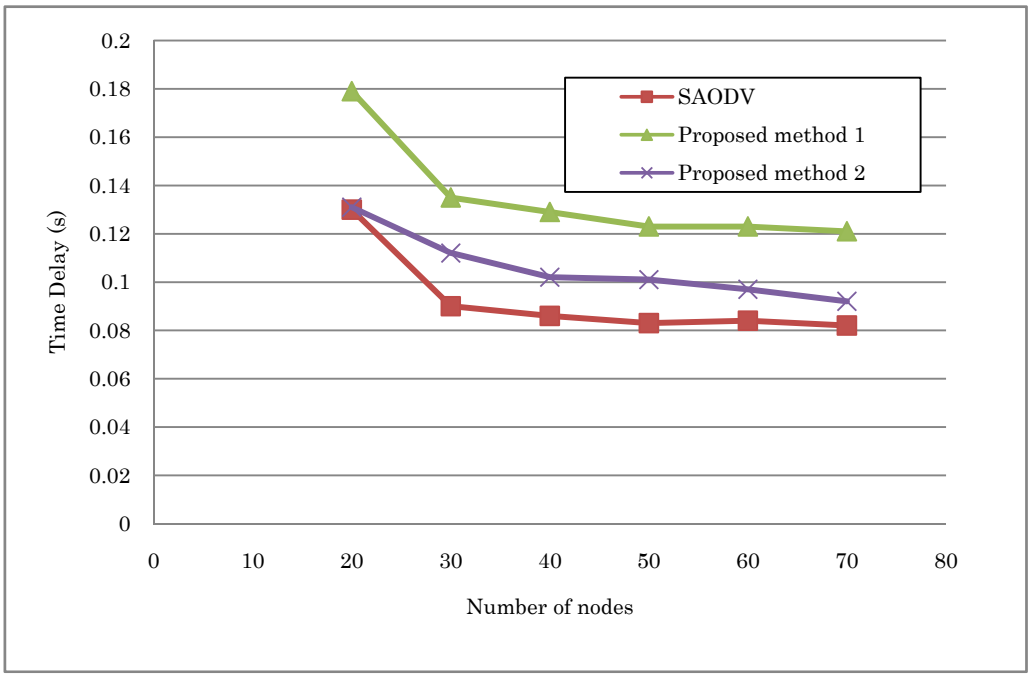


Figure 24 Time delay vs number of nodes (one malicious node)

Figure 15 shows the probability distribution of the number of hops between the source node and the destination node. We define the concept of "hop" as follows: if node A can communicate with node B directly, then, there is one hop between them. From Figure 15, following the simulation scenario shown in Table 3, we can learn that most of the cases in our simulation are of three hops, which means there are two nodes between the source node and the destination node.

In our simulation, we made every node except node 0 and node 1 (we set these two nodes as the source node and the destination node as described in section 5.3) first send RREQ message to a randomly selected node in a time interval of 10s. Therefore, when the source node 0 begins to broadcasts a RREQ message to the destination node 1 at the time of 150s, the RREP sender exists. Figures 16 and 17 show the concrete number of RREP senders between the source node and the destination node.

Before evaluating the time delay of our method, we first consider the detection time which is a part of time delay when there is a malicious node in the route. The detection time is the time taken from the point time of the broadcasting of RREQ message to the point time of the detection of a malicious node in the route. In our method, there mainly exist three kinds of malicious behavior:

1. Traditional black hole attack (SN spoofing)

2. SREQ dropping

3. SREP changing

We believe that each of these malicious behaviors has a different impact on the value of the detection time and the results are shown from Figure 18 to 20.

Figure 18 shows the detection time of SREQ dropping. In this graph, it shows that the detection time of method is less than the previous method, which means, if a malicious node plays the attack of SREQ dropping, this malicious node will be detected immediately.

Figure 19 and 20 shows the detection time of black hole attack and SREP changing. At this time, our method spends more time than the previous method because in our method, the source node should wait for the arriving of SREP message.

Figure 21 demonstrates the impact of mobility speed of nodes on time delay when there is no malicious node in the selected route. The time delay increases a little when the mobility speed increases because of more link breakdowns. If there is no malicious node, the time delay of the proposed method 2 is almost the same as the original AODV and is smaller than that of the proposed method 1, since the source node with method 1 should wait for the arrival of SREP even when there is no malicious node in the selected route. Since SAODV waits for all of the RREPs and makes comparison among these RREPs, the time delay of SAODV is a little larger than those of AODV and the proposed method 2.

In Figure 22, AODV is not shown since if there is a malicious node in a route, very few of the data packets can reach the destination node. Both of our methods have a larger time delay than SAODV since they need more time to find out a secure route. The proposed method 2 needs less time than the proposed method 1, because the intermediate node can reselect a new route.

Figure 23 and figure 24 show the impact of the total number of nodes on the end-to-end delay. First, the time delay does not change much when the number of nodes increases because the source and destination pairs of all CBR applications are the same for each scenario regardless of the number of nodes. But for less number of nodes, all protocols take more time since alternative routes are limited. Second, our methods have a little more time delay than others since our methods take more time to find out a secure route.

5.4 False positive rate

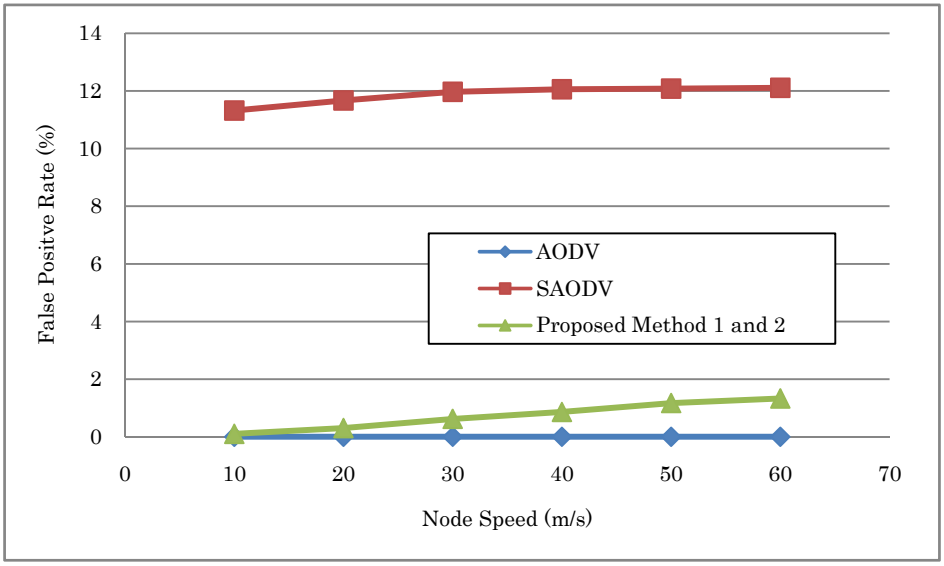


Figure 25 False positive rate vs node speed

As shown in Figure 25, the false positive rate of our proposed method increases a little along with the increase in the node speed. The reason is as follows. Consider a case where nodes A and B have a transmission range of R, and assume both nodes have a low speed v (m/s) and moves randomly. In this case, node A, after having received RREP message generated by node B, node A will wait for the SREQ message within time T. The distance that node B moves during the time T will be vT (m). However, if node B raises its speed to V (m/s), the distance will be VT (m), which means that the possibility that node B moves out of the transmission range of node A and therefore node A cannot receive the SREQ message in time T and makes a misjudgment becomes larger. From Figure 18, the false positive rate of our method is much lower than that of the previous method AODV. The reason is as follows. In the previous method, every node in the network can calculate the average value of SNs and this value is used for the judgment. Therefore, even if a node generates a RREP with a correct SN, this node can still be misjudged as malicious by those nodes that do not update the average SN value in time. On the contrary, both of our methods always use the correct SN generated by the destination node, other normal nodes only play a role of monitor to ensure that the correct SN can be transferred to the source node safely, and the monitor function entirely avoids the misjudgment that can be performed in the previous method.

5.5 False negative rate

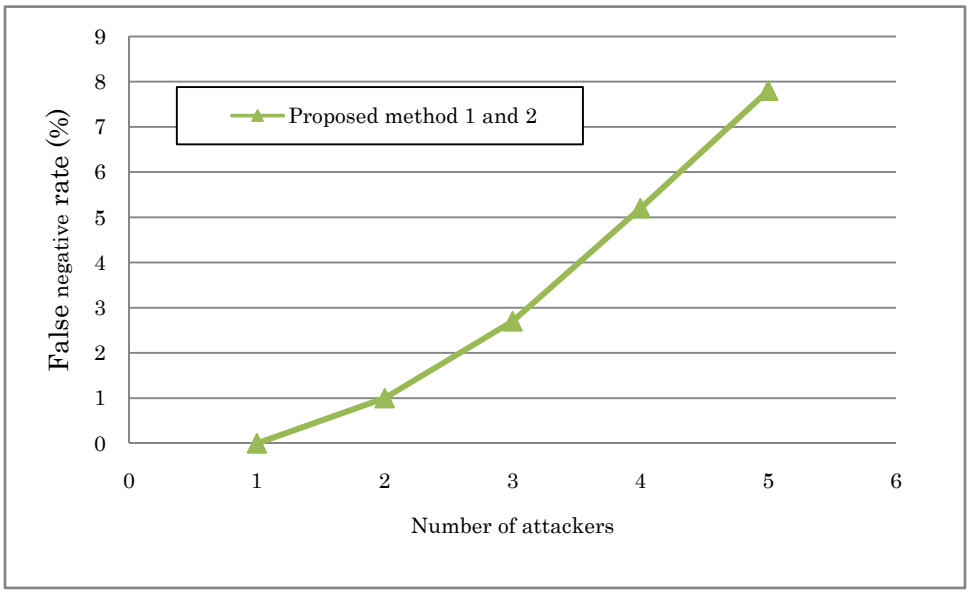


Figure 26 False negative rate vs number of attackers

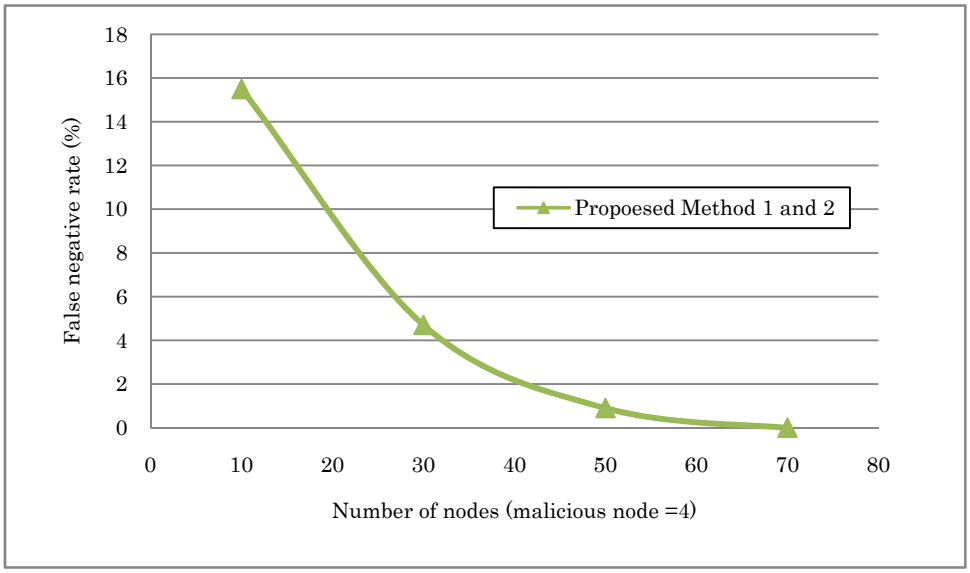


Figure 27 False negative rate vs number of nodes (malicious node=4)

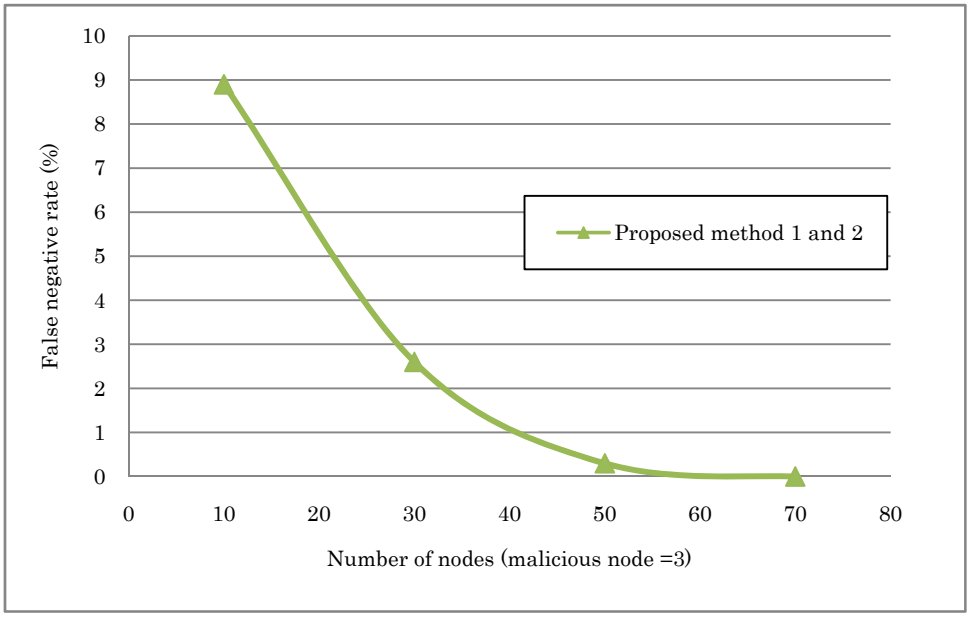


Figure 28 False negative rate vs number of nodes (malicious node=3)

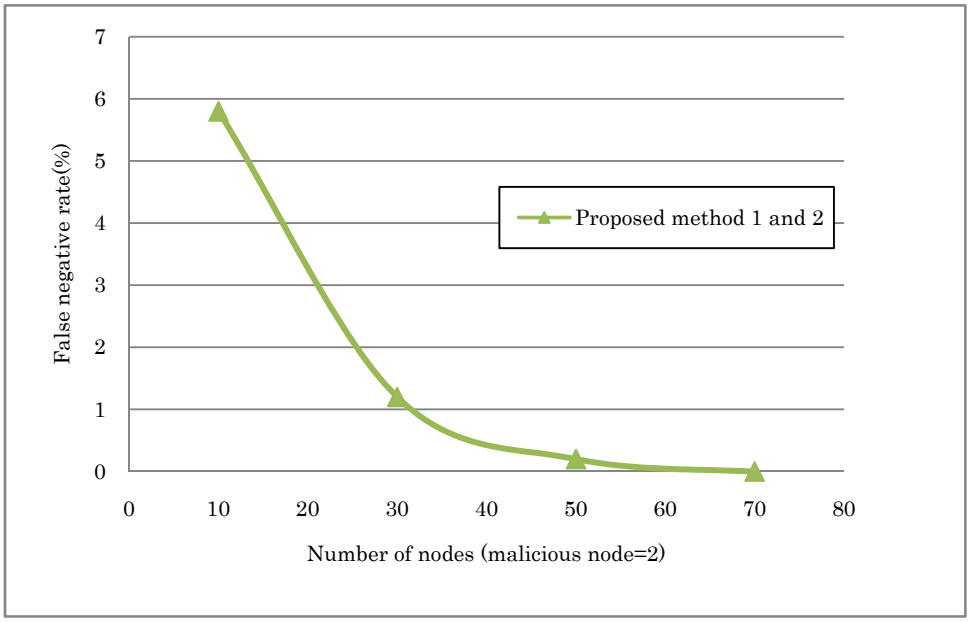


Figure 29 False negative rate vs number of nodes (malicious node=2)

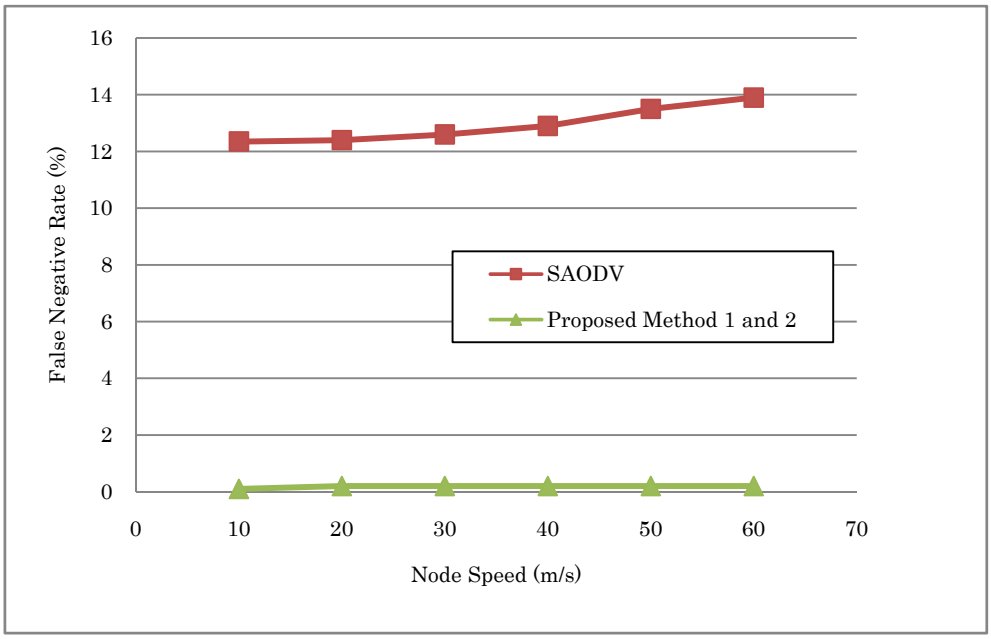


Figure 30 False negative vs node speed (malicious node=1)

As shown in figure 26, when the number of malicious nodes or attacker is 1, the false negative rate is very close to 0. Although the rate increases along with the increase in the number of malicious nodes, the false negative rate is still less than 8% even if 17% of the nodes are malicious.

Figure 27-29 show relationship between the density of the network and the false negative rate. It is clear that with the increase of the node number, the false negative decrease sharply. In other words, the detection rate increases nearly to 100%. The reason of the decrease of the false negative is that, in our method, we need other nodes to play a role of monitor in order to monitor the malicious behavior such as SREQ dropping and SREP changing. If there are more nodes in the network, there will be more monitor in the network. Therefore, more malicious nodes will be detected.

In order to show the effectiveness of our method in the aspect of false negative rate, we compare the false negative rate of our proposed methods with that of the above mentioned previous method based on dynamic learning [13]. Figure 25 shows the results of such comparative simulation.

In our methods, a source node believes that the SN in the SREP generated by the destination node is the largest and on the reception of SN, this source node can make a decision on whether there is a malicious node in the route or not. Therefore, if a malicious node generates a faked RREP, this malicious node will be detected with the help of monitor function with nearly 100% probability as shown in Figure 20.

5.6 Control packet overhead

Control packet overhead is denoted by O and defined as follows:

$$O=\left(\sum_{i=1}^{n} Ci + \sum_{j=1}^{m} Cj + \cdots + \sum_{k=l}^{l} Ck\right) \times Csize$$

Where Ci, Cj…Ck denote control packets in the method, and n, m…l denote the total number hops of transmission of each control packet. Csize is the size of each control packet. In this simulation, the size of all the control packets is assumed 24 byte.

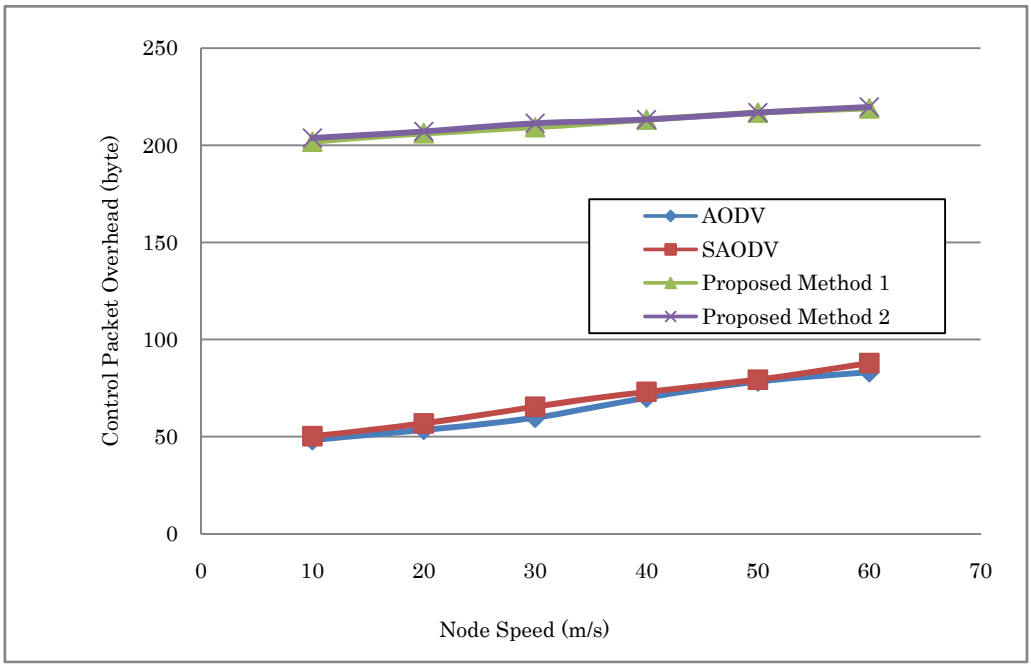


Figure 31 Overhead vs node speed (no malicious node)

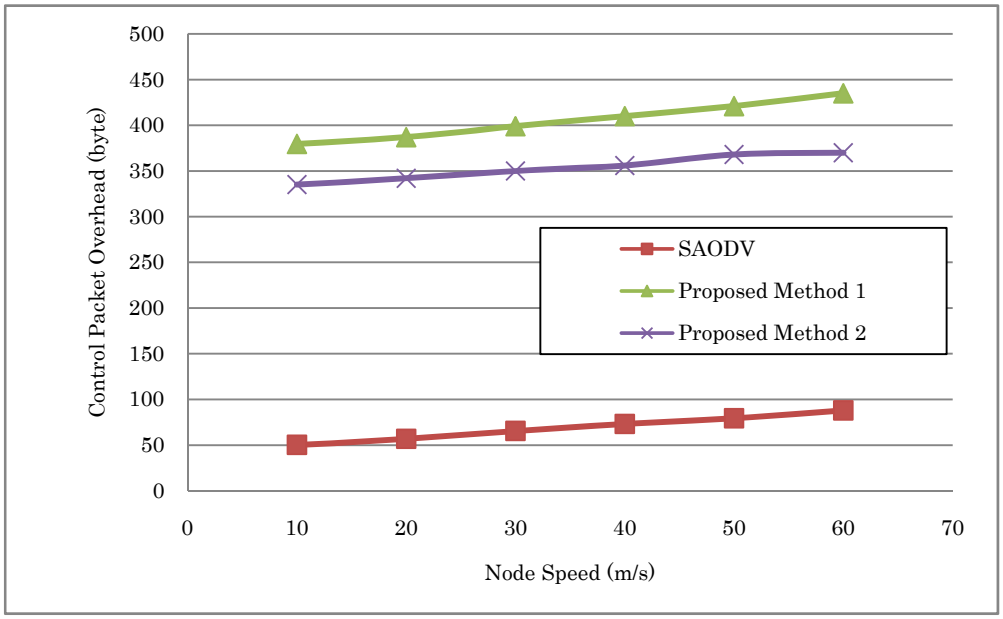


Figure 32 Overhead vs node speed (one malicious node)

Figures 31 and 32 illustrate the impact of the mobility speed of nodes on control packet overhead. First, in all the methods, the overhead becomes larger with the increase in the mobility speed of nodes. Since higher speed of nodes leads to more link breakdowns which will result in more route discoveries, it will increase the number of requests and introduce more overhead. Second, AODV has 50 to 87 byte overhead when there are no black hole nodes as shown in Figure 31 and SAODV has almost the same amount of overhead as the AODV, since SAODV does not use any newly defined control packet. Since our methods use more route requests and SN requests to check the SN, our methods have 6-7 times more overhead than the other two methods as shown in Figure 32. Figure 32 shows the results when there is a malicious node in the selected route. The overhead of our proposed methods increases by 6-9 % since more route request packets will be sent after this malicious node is detected. The reason why the proposed method 2 has less overhead than the proposed method 1 is the fact that the intermediate node in the route can select a new route by the method 2, which makes smaller the number of nodes forwarding the route request messages.

6. Conclusion

In this paper, we proposed a new destination SN based method which can efficiently detect the black hole attack in MANET. In our method, by checking the reply messages sent by the destination and the intermediate nodes, the source node can detect black hole attackers. Meanwhile, if a normal node can monitor the messages relayed by the intermediate nodes in the route between the source and the destination nodes, then the normal node can detect the cooperation of the malicious nodes in the route and inform the source node of the detection results. After the detection of the black attackers, the source node will retry to set up a route without attackers. The simulation results have demonstrated that our method shows significant effectiveness in detecting the black hole attack and has more control overhead of route request and time delay.

7. Future work

In the future, we will mainly solve two problems related to the proposed method. The first problem is what we call false alarm message broadcasting that is shown as follows. In our proposed method, if a monitor detects that there exists a malicious node, this monitor broadcasts an alarm message containing this malicious node's ID to inform the source node and other nodes that there exists a malicious node in the route. However, if this monitor is malicious, it is entirely possible that this malicious monitor broadcasts a false alarm message to libel other nodes as black hole attackers.

In our proposed method, we define two kinds of alarm messages. The first kind of alarm message is broadcasted after detecting a SREQ dropping and the second one is broadcasted after detecting a SREP changing. Accordingly, there are two kinds of false alarm message. In order to avoid these two kinds of false alarm message, we design the following scheme: as for the SREQ dropping alarm, suppose that node A broadcasts an alarm message to claim that node A's neighbor node B does not forward the SREQ message to the destination node, if node B broadcasts the SREQ message and node B can overhear this alarm message at the same time, this alarm message should be a false alarm message. At this time, node B also broadcasts an anti-alarm message. In this anti-alarm message, node A's ID and the contents of the alarm message broadcasted by node A should be contained. All of the neighbors of node B can overhear the SREQ message since node B has indeed forwarded this message, and therefore, after receiving this anti-alarm message, these neighbor nodes can make a judgment whether node B is innocent or not. Then, these neighbor nodes unicast their judgment reports to the source node. The source node collects all of the report from various nodes and makes a decision. As for the SREP message changing alarm, the theory is the same as what was described above.

However, we should also consider the cases where some neighbor nodes do not want to make such a judgment for node B. What is worse, some neighbor nodes might be the accomplices of node A. All of these cases may make the source node to make a misjudgment and leads to the increase in the false positive rate and false negative rate of our proposed method. Therefore, we will improve the current method and evaluate the effectiveness of this improvement in the future.

The second problem is what we call SN collector problem. SN collector is defined as follows. Since the SN generated by a destination node is very important, a malicious node can play a role of SN collector in order to get the SN of as many other nodes as possible by broadcasting RREQs with high frequency to different nodes in a MANET so that this collector always keeps the freshest SN of other nodes. If this malicious node wants to play the black hole attack, this node can just send back the RREP with the up-to-date SN it has collected and does not need to drop the SREQ message nor change the contents of the SREP

message. Therefore, with our current monitor scheme, it is very hard to detect the SN collector. Therefore, how to solve this problem is our next issue.

In our method, we use the monitor in our detection schemes, taking advantage of its ability to provide first-hand and direct observations of the nearby traffic which means if node A is within the range of a node B, node A can overhear communications to and from B even if those communications are not directed to A. However, we should also consider that the limited capacity offered by the mobile terminals makes this approach expensive as it captures every packet associated to itself and its neighbors. Moreover, data from promiscuous monitoring can be unreliable under various conditions such as the link breakdown or failure. Therefore, in the future, we should consider the loss of packets as a metrics to evaluate our proposed method.

References

[1] C.E.Perkins, E.M.B. Royer and S.R.Das, “Ad Hoc On-Demand Distance Vector (AODV) routing”, RFC 3561, July 2003.

[2] M. Hollick, J. Schmitt, C.Seipl and R.Steinmetz, “The ad hoc ondemand distance vector protocol: an analytical model of the route acquisition process”, Proc. of Second Intl Conference on Wired/Wireless Internet Communications (WWIC'04), Frankfurt, Feb 2004, pp. 201-212.

[3] M. Hollick, J. Schmitt, C. Seipl and R.Steinmetz, “On the effect of node misbehavior in ad hoc networks”, Proc. Of IEEE Intl Conference on Communications (ICC'04), Paris, June 2004, pp. 3759-3763.

[4] I. Stamouli, P. G. Argyroudis and H. Tewari, “Real-time intrusion detection for ad hoc Networks”, Sixth IEEE Intl Symposium on a World of Wireless Mobile and Multimedia Networks (WoWMoM'05), 2005, pp. 374-380.

[5] B. Sun, Y. Guan, J. Chen and U. W.Pooch, “Detecting black-hole attack in mobile ad hoc networks”, Proc. 5th European Personal Mobile Communications Conference, Apr 2003, pp. 490-495.

[6] Y.A. Huang and W.Lee, “Attack analysis and detection for ad hoc routing protocols”, 7th International Symposium on Recent Advances in Intrusion Detection (RAID’04), French Riviera, Sept 2004, pp. 125-145.

[7] Y. Huang, W. Fan, W. Lee and P. Yu, “Cross-Feature analysis for detecting ad-hoc routing anomalies”, Proc. of the 23rd IEEE Intl Conference on Distributed Computing Systems (ICDCS'03), May 2003.

[8] A. K.Ghosh and A. Schwartzbard, “A study in neural networks for anomaly and misuse detection”, 8th USENIX Security Symposium, 1999.

[9] W. Lee, S.J. Stolfo and K.W. Mok, “A data mining framework for building intrusion detection models”, IEEE Symposium on Security and Privacy, 1999, pp. 120-139.

[10] W.W. Cohen, “Fast effective rule induction”, Machine Learning: the 12th International Conference, Lake Taho, CA, 1995.

[11] J.R. Quinlan, “C4.5: Programs for machine learning”, Morgan Kaufmann, San Mateo, CA, 1993.

[12] X. Wang, T. Lin and J. Wong, “Feature selection in intrusion detection system over mobile ad-hoc network,” Technical Report, Computer Science, Iowa State University, 2005.

[13] S. Kurosawa, H. Nakayama, N. Kato, A. Jamalipour and Y. Nemoto, “Detecting blackhole attack on AODV-based mobile ad hoc networks by Dynamic Learning Method”, Intl Journal of Network Security, vol 5, no. 3, Nov. 2007, pp. 338-346.

[14] S. Marti, T. Giuli, K. Lai and M. Baker, “Mitigating routing misbehavior in mobile ad hoc networks”, Proc. of the Sixth Annual Intl Conference on Mobile Computing and Networking (MOBICOM), Boston, 2000.

[15] I. Stamouli, “Real-time intrusion detection for ad hoc networks”, Master's thesis, University of Dublin, September 2003.

[16] H. Deng, W. Li, and D. P. Agrawal: “Routing security in wireless ad hoc network”. IEEE Communications Magazine, pages 70–75, (2002)

[17] Latha Tamilselvan, V. Sankaranarayanan: “Prevention of Black Hole Attack in MANET”, The 2nd international conference on wireless, Broadband and Ultra Wideband Communications (January 2007)

[18] Mohanmmad Al-Shurman et al: “Black Hole Attack in Mobile Ad Hoc Network”, ACMSE’ 04, (April 2004)

[19] S. Ramaswamy, H. Fu, M. Sreekantaradhya, J. Dixon, and K. Nygard: “Prevention of cooperative black hole attack in wireless ad hoc networks”, Proceedings of 2003 International Conference on Wireless Networks (ICWN’03), pages 570–575. Las Vegas, Nevada, USA, (2003)

[20] I. Rubin, A. Behzad, R. Zhang, H. Luo, and E. Caballero. Tbone: “A mobile-backbone protocol for ad hoc wireless networks” , In Proceedings of IEEE Aerospace Conference, volume 6, pages 2727–2740, (2002)

[21] The VINT Project, UC Berkeley, LBL, USC/ISI, and Xerox PARC. “The ns Manual”, 2005

[22] University of California and Lawrence Berkeley Laboratory, AODV source code for network simulator, (1997)

[23] Network Simulator Official Site for Package Distribution, web reference, http://www.isi.edu/nsnam

Publications

1. XiaoYang Zhang and Yasushi Wakahara: “Sub-Marine Attack and its Defense in Ad Hoc Network Routing Protocols” IEICE Technical Committee on Information Network (IN). (January 2008)

2. XiaoYang Zhang and Yasushi Wakahara: “Detecting Sub-Marine Attack with Safe-triangle and GPS in Ad hoc Network”, IPSJ General Conference 2008. (March 2008)

3. XiaoYang Zhang, Yuji Sekiya and Yasushi Wakahara: “Defending Ad Hoc Network from Black Hole Attack”, IEICE Society Conference 2008 (September 2008)

4. XiaoYang Zhang, Yuji Sekiya and Yasushi Wakahara: A method of detecting black hole attack in mobile ad hoc network”, the 5th Technical Committee on Network Software, (November 2008)

5. XiaoYang Zhang, Yuji Sekiya and Yasushi Wakahara: “Detection of the Black Hole Attack in a Mobile Ad hoc Network (MANET)” the 6th Technical Committee on Network Software. (February 2009)

6. XiaoYang Zhang, Yuji Sekiya and Yasushi Wakahara: “Evaluation of Methods to Detect Black Hole Attack in MANET”, IEICE General Conference 2009. (March 2009)

7. XiaoYang Zhang, Yuji Sekiya and Yasushi Wakahara: “Proposal of a Method to Detect Black Hole Attack in MANET”, the 9th International Symposium on Autonomous Decentralized Systems. (ISADS 2009)