Hindawi
Mobile Information Systems
Volume 2017, Article ID 8636379, 13 pages
https://doi.org/10.1155/2017/8636379

Research Article

Efficient Hybrid Detection of Node Replication Attacks in Mobile Sensor Networks

Ze Wang, Chang Zhou, and Yiran Liu

School of Computer Science and Software, Tianjin Polytechnic University, Tianjin, China

Correspondence should be addressed to Chang Zhou; zhouc1993@sina.com

Received 5 February 2017; Revised 16 May 2017; Accepted 14 June 2017; Published 21 August 2017

Academic Editor: Massimo Condoluci

The node replication attack is one of the notorious attacks that can be easily launched by adversaries in wireless sensor networks. A lot of literatures have studied mitigating the node replication attack in static wireless sensor networks. However, it is more difficult to detect the replicas in mobile sensor networks because of their node mobility. Considering the limitations of centralized detection schemes for static wireless sensor networks, a few distributed solutions have been recently proposed. Some existing schemes identified replicated attacks by sensing mobile nodes with identical ID but different locations. To facilitate the discovery of contradictory conflicts, we propose a hybrid local and global detection method. The local detection is performed in a local area smaller than the whole deployed area to improve the meeting probability of contradictory nodes, while the distant replicated nodes in larger area can also be efficiently detected by the global detection. The complementary two levels of detection achieve quick discovery by searching of the replicas with reasonable overhead.

1. Introduction

Wireless sensor networks (WSNs) typically consist of large numbers of sensing devices to collect information in a monitored field. Such networks have been used in many fields, including object tracking, disaster rescue, environmental monitoring, and coverage calculation. As WSNs could be deployed in hostile environments, the networks can be unsupervised and the node hardware is generally not tamper-resistant; kinds of notorious attacks would be suffered. An application-independent intrusion known as the node replication/clone attack, works by replicating or cloning nodes with the copied node ID and all cryptographic materials associated with that ID. In this case, adversaries could compromise one node and fabricate many replicas with the same ID of the captured node. Then, these nodes will be deployed back into the networks to eavesdrop communications or launch attacks. The code implanted into a rogue replica can yield normal communication with other nodes without being easily detected.

Recently, due to advances in robotics, mobile sensor networks (MSNs) have become feasible and applicable. The clone attacks on sensor nodes are harder to be defeated in mobile sensor networks than in static one. Although the problem of node replica detection in static networks has been extensively studied [1–5], only a few schemes have been proposed for mobile sensor networks [6–9].

While the clone detection problem has been recently addressed for mobile sensor networks, the problem is still open when some particular mobile WSN is considered. The common idea, such as the schemes in [8, 9], is based on the meetings of a witness node with more than one of the replicated nodes or their neighbors. However, one can only count on meetings in whole deployed areas. For situations where such meetings are very difficult, for example, large areas or nodes moving quite slowly, these detection schemes may perform unsatisfactorily. Therefore, we deem that there are still spaces to further facilitate the meetings of the witness nodes or the meetings of the contradictory location claims.

Our scheme constructed a distributed hybrid algorithm—a two-dimension detection scheme, combining local detection and global detection. By adding the global detection mechanism, we solve the above problems and focus on solving the problem that witness cannot meet the replicated node timely. Global detection enables the location message of the replicated node to be transmitted to

close position from a distant location, then speeding up the encounter of the paradox location messages. Thus, detection result is performed better. When nodes move faster, the local detection mechanism will play a major role in detecting contradictory location messages. When nodes move at a slow speed, the global detection mechanism solves the encounter problem of the paradox location messages.

Contribution. (1) In this paper, we propose a novel distributed replica detection scheme for MSNs, in which the detection process is carried out at two levels. Through numeric analysis, we find that the proposed scheme improves detection probability in acceptable time intervals.

(2) Our hybrid detection scheme combines the advantages of static network detection protocol and node collaboration. When the node speed is close to zero, we can still rely on the global detection to detect the replicated nodes efficiently.

(3) Hybrid detection protocol can be applied to more replicated node distributions (including the situation where replicated nodes are far away from each other), which has higher adaptability than the existing algorithms.

(4) While improving the detection rate, our scheme does not add extra overhead. Analysis and simulation results support our findings even when there is only one clone in the network, which is the worst case for replica detection.

Paper Organization. The rest of the paper is organized as follows. Section 2 reviews the related work. In Section 3, we introduced the network and security models. In Section 4, we detail TD, the novel replication detection scheme for MSNs, and its prime rationales. The security analysis and performance evaluation are reported in Section 5 and we followed to perform simulations whose results are described in Section 6. Section 7 presents our conclusions.

2. Related Work

Xing and Cheng [7] have proposed a distributed scheme for detecting replication attack with location information. Every node in the network must exchange its record list with neighbors to avoid illogically behaved replicas. That is, each node plays two roles, as a normal node and a witness node. In [7] composed of TDD and SDD, node position has been utilized to detect replication attacks in dynamic networks successfully. Yu et al. [8] have designed a localized mechanism, including XED and EDD, to defend against node replication attacks. It has proved a memory overhead comparable to [7]. However, the scheme only withstands against node replication attack in dynamic networks. Conti et al. [9] have proposed a distributed protocol HOP that leverages only one-hop communication and node mobility to enforce the emergent property of replica detection. Although HOP is testified by experiments to be more efficient than XED, we find that the deployed area may be too large to implement the detection in fewer rounds.

A surge of security schemes have been sparkled to resist malicious attacks in sensor networks. To defeat node replication attacks, most of the existing detection schemes adopt the witness-finding strategy to collect discrepancy messages. When there are replicas in the network, the witness nodes, according to the received location claims, have the possibility of finding one node ID in two distant locations, which implies that node ID is being used by replicas. Afterward, detected replicas can be excluded using network-wide revocation with the conflicting locations proof. However, when a sensor node has mobility, the same node may locate in different positions. Schemes designed for the static sensor network are not effective in the mobile environment.

To detect replica attacks in mobile ad hoc networks, Xing and Cheng [7] proposed judging rules for consistent locations of mobile nodes. The linear distance between location i and location j of the same node in different times t_i and t_j must be less than or equal to the maximum possible distance the node can traverse during the time interval $|t_j - t_i|$.

A general detection framework, called local information exchange proposed in [7], is that whenever two nodes meet each other, they exchange authenticated messages including identity, time, location, the lowest-order unused key in a one-way hash chain, and signature.

Messages from different nodes should have consistent location and challenge claims of a node u according to the location check lemmas elaborated in [7]. Any violation of these rules signals a node replication attack on identity u. In Time-Domain Detection (TDD), neighbors of a node u will send the messages received from u to the same location in the network. The node closest to that location is responsible for receiving these reports and launching a check on them. In Space-Domain Detection (SDD), each node maintains a table which records the information received from the nodes it encountered within the past time units. Once some of the witness nodes meet each other and exchange their recorded information about identity u, they may find contradictory information breaking the lemmas. However, this scheme requires each node to store a set of messages for every monitored node and an inverted hash chain. The storage overhead may be not viable for a sensor node.

Ho et al. [6] proposed a centralized detection algorithm for mobile sensor networks using Sequential Probability Ratio Test (SPRT). Intuitively, by having each node send the location of each encountered node, the base station can check if there is a node appearing at two distant locations with a velocity exceeding the predefined limit. If such a node exists, it is very likely to be a replica. Nevertheless, practically there could be some errors in the node speed measurement, leading to either false positives or false negatives. To avoid false decisions, SPRT uses specific sequential hypothesis test with null and alternative hypotheses corresponding to normal and replica nodes to determine which hypothesis should be accepted with the consideration of a sequence of observations. However, SPRT relies on the involvement of the base stations, easily incurring problems like single-point failure and fast energy depletion of the sensor nodes around the base station.

Yu et al. [8] proposed two fully distributed schemes XED and EDD. In XED, two nodes exchange a random number when they meet. At their next meeting, they check if the random numbers they are storing are the same and eventually exchange a new random number for the next

meeting. This protocol is simple and seems to be very suitable for resource constrained devices like sensor nodes. In fact, the storage requirement is limited as well as the required computations. However, the replicas may collude to defeat XED by sharing the random number through some hidden communication channels. EDD is proposed to mitigate the collusion attack. The idea behind EDD is motivated by the following observations. The number of times that a specific node is encountered during a fixed period of time should be within maximum and minimum probability thresholds. According to these observations, if each node can discriminate between these two cases, it has the ability to identify the replicas. The two thresholds used for replica detection may be infected by parameters like node density, moving velocity, and mobility pattern; thus inaccuracy parameters used in the offline computation process of EDD may lead to false decisions of replica detection.

Conti et al. [9] proposed a concise distributed replica detection scheme HIP/HOP where time is divided into rounds. Sensor nodes broadcast their location claims to perform location information exchange in every round. i rounds of location claims are stored into history logs. Neighboring nodes exchange and compare their history logs. The two proposed protocols, HIP and HOP, differ in how they perform this comparison. In particular, in both protocols, each node compares its own log with the logs received from its neighbors. However, in HOP each node compares with not only the history logs of encountered nodes but also their received logs. Comparing history logs with just direct neighbors only requires one-hop communications: information related to sensors distant more than one-hop away will be implicitly spread in the network by moving sensors. Although the basic idea of finding location inconsistency is nearly the same as SDD, HIP/HOP has the advantages over SDD on concise implementation and less storage requirement.

Many previous detection protocols for mobile networks are based on the mobility of nodes and the mutual cooperation between nodes, which leads to lower detection efficiency when the nodes are moving at low speed. Furthermore, the detection scheme of the static sensor network can not meet the requirements of the mobile network. This paper is inspired by the distributed replica detection scheme HIP/HOP [9] and our preliminary work [10]. We extend our preliminary work in many directions: we incorporated the global detection algorithm and considered different type of adversary. In order to clearly measure our experiment, we compared the results between HIP/HOP and TD under different conditions in Section 6.

3. System Model

3.1. Network Model. We assume the MSN is composed of N sensor nodes with IDs : $ID_1, \ldots, ID_N$. We denote with n_i the c replicated node and assume that there are c replicas of n_i in network, meaning c replicas with the same ID of n_i. Each node has a communication radius R and the communication is assumed to be symmetric. The time is divided into time epochs, each with same length Δ. We assume that sensors rely on loose time synchronization. The neighboring nodes exchange messages in the middle of the time interval and loose time synchronization would ensure that the nodes have synchronized time interval counts. Suppose the nodes are randomly and uniformly deployed in the network and their geographic positions are known by GPS or other position-awareness services. The network provides geographic routing to enable message transmission toward a specific location in the network. The sensor nodes move according to the Random WayPoint (RWP) model [11], which is commonly used in modeling the mobility of ad hoc and sensor networks. The moving speed of the nodes has an upper limit $v_{\max}$. Motions are characterized by two features: (1) the maximum speed and (2) the pause time. During simulation each node starts moving from its initial position to a random object point, selected inside the simulation area. The motion speed is uniformly distributed between zero and a maximum speed. When a node reaches the target point, it waits for the pause time and after that, by selecting another random target point, it moves again.

3.2. Security Model. In accordance with the existing works, we assume that all nodes in the network are not tamper-resistant, and the compromise or capture of a node releases all its security material to the attacker. An adversary that has captured a node k can deploy replicas of the node anywhere in the network. Note that the replicas own all the legitimate information of the compromised node (e.g., ID, keys, and code). Thus they can easily participate in network operations acting same as legitimate nodes and therefore launch various internal attacks afterward. For example, a replica can fabricate messages to mislead the decision makers or keep injecting bogus data to cause a network outage. Because this paper focuses on the considered node replication attack, other attack scenarios such as key managements, replay attacks, wormhole attacks, and Sybil attacks are not discussed here.

Furthermore, we utilize the identity-based public key system [12] to enable authentication of nodes similar to some of the existing works, so identity-based signature generation and verification are feasible. Let $Sig_k(M)$ denote the signature to the one-way hash value of the message M with the private key of node k. Owing to the use of the signatures, the replicas cannot create a new ID or disguise themselves as nodes never compromised before, because it is too difficult for adversaries to get the corresponding security verification. The nodes in the system can verify the signature with the ID of node k and the public key easily.

The encoding method uses a one-way hash function MD5 to be encrypted plaintext "summary" into a string of 128-bit ciphertext, this string of ciphertext is also known as digital fingerprint (fingerprint), it has a fixed length, and different plaintext abstracts are consistent so that this string of abstracts can make it a "fingerprint" that verifies whether the plaintext is "true."

3.3. Notation. For better understanding of the proposed scheme, we give the main notation defined in the scheme.

sn: the sequence number of the epochs.

f: the quantity of the slots in an epoch.

t_k: the sequence number of the current slot in the current epoch, $t_k \in [0, \ldots, f-1]$.

$LocClaim_{k,p}$: the location claim of node k in the slot p of the current epoch.

$LocCert_{k,sn}$: the location certificate used to indicate a representative location in an epoch for the global detection, equal to $LocClaim_{k,T_k}$.

T_k: the sequence number of the representative slot in an epoch.

l_k, loc_k: the node location in a time slot.

n_d: the quantity of the neighbor nodes.

$ClaimList_j[i]$: the array for node j storing $LocClaim$s received at time slot i.

$VLoc_k$: the global detection location which the $LocCert$ of the node k is sent to.

$VReq_k$: the request packet for the verification of the $LocCert$ of the node k

$VList$: the array of storing $LocCert$s for the global detection.

4. The Proposed Scheme

4.1. Overview. To facilitate the discovery of contradictory location claims, it would be preferable that the witness nodes are distributed in some local area. However, if a node u and its replica u' are far away from each other, their witness nodes who have received the location claims of the replicas may take a long time or a long journey to run into each other. One method to force the contradictory location claims to meet earlier is to send the location claims including the same ID, no matter coming from a single node or replicas of the same node, to some checking location in the network. When a location claim reaches the checking location, the monitored node may have moved to a new location. The location in the location claim message is only a representative of node locations in a period with time length tl; the actual position of the node should be located in a circular area with radius length of $v_{\max} * tl$. Noting that replica nodes locating in the same circular area may escape from the above detection, local information exchange would be employed for replica detection in the local area.

Owning to the above motivation, we propose two levels of detection to improve the meeting probability of the location claims including the same ID. Two levels of time units are introduced to regulate the detection cycles. Epoch is used as the coarse grain time unit. In every epoch, messages including representative location of the nodes with the same ID are sent to a position for global detection. Slot is used as the fine grain time unit. In every slot, the neighboring nodes exchange location messages for local detection. Since the local detection would be performed in a local area much smaller than the whole deployed area, the meeting probability for the witness nodes of different replicas is expected to increase obviously.

Time is divided into equivalent epochs. The length of the epoch is Δ and the sequence number of an epoch is denoted by sn. In an epoch, the maximum distance a node can travel is $ML = v_{\max} * \Delta$. An epoch is divided into f time rounds/slots. The neighbor nodes exchange their location claims and then carry out location verification in every slot. The replica detection is performed by the check of the accordance of the location claims of the same node in time and space. In some specific slot of some specific epoch, locations claimed from the same node should be not conflict, or there must exist a replication attack to the node.

The proposed replica detection scheme is composed of two levels of detection, the local detection and the global detection. In every epoch, a node should generate a new $LocCert$ for the global detection. In every slot of an epoch, a node should broadcast its $LocClaim$ for local detection. Therefore, two levels of accordance checks in time and space are performed based on the $LocCert$ and the $LocClaim$, respectively.

We give the definition of the location certificate as

$$LocCert_{k,sn} \coloneqq \langle ID_k, sn, T_k, loc_{k,sn}, Sig_k\left(H_{MD5}\left(ID_k \parallel sn \parallel T_k \parallel loc_{k,sn}\right)\right)\rangle. \quad (1)$$

Therein, sn is the sequence number of the epoch. T_k is the number of the slots when the certificate is generated. $loc_{k,sn}$ denotes the claimed location of node k in the T_k slot of the sn epoch. For the balance of overhead in time axis, T_k should be uniformly distributed in a region $[0, f-1]$. One of the T_k selection methods is given by $T_k = g(ID_k) = (ID_k + C) \bmod f$, where C is a constant.

In every slot, a node k generates its location claim as

$$LocClaim_{k,t_k} \coloneqq \langle ID_k, sn, t_k, l_k, Sig_k\left(H_{MD5}\left(ID_k \parallel sn \parallel t_k \parallel l_k\right)\right)\rangle. \quad (2)$$

Therein, t_k is the number of the slots and l_k is the location of the node. The neighbor nodes exchange the $LocClaim$ and $LocCert$, which are called location claim exchange, to launch a detection procedure (see Algorithms 1 and 2).

4.2. Two Levels of Detection. By receiving the location claim of a neighbor k, the node j executes the following local detection procedure.

When receiving $LocClaim_k$ from a neighbor node, the following steps will not be executed if the node ID is invalid or verification to the signature fails. Thus DoS attacks from illegal nodes are alleviated. Next, if the claimed location is out of the communication range, node j will send an alarm message to report the abnormal location. We enlarge the communication range threshold by introduction of parameter $\gamma \in [0, 1)$ to decrease false negatives. A node will be put into the black list if it is accused multiple times by abnormal location alarm messages. For every one of the latest f slots, the node will compare its own $ClaimList[i]$ with the neighbor's. Node replication attack will be identified by contradictory location claims.

Transmission of the location certificates for global detection includes the following steps.

Function $LocMap(ID_k, sn)$ is used to uniformly map ID_k, $k \in [1, N]$, to some location in the deployment region.

```
(1) if Verify-signature(LocClaim_k)== false then
(2)    return;
(3) end if
(4) if |l_k − l_j| > (1 + γ)R then
(5)    Raise-alarm(LocClaim_k, LocAbnormal);
(6)    return;
(7) end if
(8) Store LocClaim_k into ClaimList_j[t_k];
(9) for each i ∈ [0, f − 1] do
(10)    Compare ClaimList_k[i] with ClaimList_j[i];
(11)    for each pair of contradictory LocClaims on node u do
(12)       Raise-alert(LocClaim_u, LocClaim_u');
(13)    end for
(14) end for
```

Algorithm 1: Local detection algorithm.

```
(1) Set VCert_k = 0 and compute T_k = g(ID_k);
(2) if t_j == T_k then
(3)    Set VCert_k = 1 with a probability p_v = 1/d;
(4) else
(5)    if LocClaim_{k,T_k} ∉ ClaimList_k[T_k] && (t_j − T_k) mod f ≤ 2 then
(6)       Set VCert_k = 1;
(7)    end if
(8) end if
(9) if VCert_k == 1 then
(10)   Compute VLoc_k = LocMap(ID_k, sn);
(11)   Set VReq_k = VLoc_k ‖ LocClaim_{k,t_j};
(12)   Send VReq_k to a location VLoc_k;
(13) end if
```

Algorithm 2: LocCert transmission for global detection.

One of the mapping methods is described as follows: divide the region into N subregions uniformly and let $Cp[i], i \in [1, N]$, denote the centroid coordinate of subregion i; then $LocMap(ID_k, sn) = Cp[(ID_k + sn) \bmod N]$.

Since the message $VReq_k$ could be generated from different neighbors of node k, the duplicated $VReq_k$s may incur additional communication overhead. Addressing this problem, a limited forwarding policy has been adopted. In some slot p, if a routing node j receives more than one $VReq_k$s including the same ID_k, the node j will compare the locations in $LocCert_k$s. If the locations are consistent, the node j just drops the later $VReq_k$; otherwise it would raise an alert with conflicting *LocCert*s.

When a node j receives $VReq_k$, it fetches $VLoc_k$ and $LocCert_k$ and then carries out global detection steps as Algorithm 3.

The local replica detection is accomplished by exchanging the location claims and the corresponding location verifications each time slot. The node verifies the validity of the current locations of the neighbor nodes and then exchanges and searches history lists of *LocClaim*s to find discrepant locations in the last f slots or a time interval Δ. In this time interval, the moving range of a node is a circle with radius $v_{\max} * \Delta$ centered at *loc* in *LocCert*.

During the local detection, a node may assemble its neighbors' *LocationCert* into the global detection request *VReq* at the appropriate time slot. The node sends *VReq* to some mapping location determined in the network. The discrepancy of the *LocationCert*s in the global detection phase identifies the replication attack on the corresponding node.

5. Analysis

5.1. Detection Probability. Let ML denote the maximum distance of a node moving in an epoch. Due to the restriction of the location cert, a node and its replica who share the same location cert can depart from each other at a distance of $2ML$ at most. Furthermore, the neighbors of the node and its replica can move a distance of ML at most in an epoch. Thus, we can deduce that the neighbors of the node and its replica must reside in a detection circle with radius $ML + R$ in one slot and with radius $2ML + R$ in one epoch. Inspired by the model used in HIP/HOP [9], we divide the detection circle into small regular hexagon cells with side $s = R/2$,

```
(1) if |VLoc_k − l_j| ≤ R ‖ NextHop == null then
(2)   for each LocCert_k' in VList do
(3)     if LocCert_k is inconsistent with LocCert_k' then
(4)       Raise-alert(LocClaim_k, LocClaim_k');
(5)     end if
(6)   end for
(7)   Add LocCert_k into VList;
(8) else
(9)   Forwarding VReq_k;
(10) end if
(11) Remove LocCerts 2 epoches older than the current time from VList.
```

Algorithm 3: Global detection.

which ensures that two nodes located in the same cell can hear each other and exchange messages. Thus, once one of the neighbors of the node meets one of the neighbors of the replica, the replication will be successfully detected. In the following, we will give some analysis on the probability of such detection.

The number of such cells in the detection circle can be computed by $q = f(ML, R)$. Let a and a' denote a pair of replicated nodes. They share common IDs, key materials, and history records except the current location claims since the pair reside in different locations for message collection. Given some slot t_i, suppose a located in l has a set of d neighbors denoted by P and a' located in l' has a set of d' neighbors denoted by Q.

Let the event C_i represent "exactly i different cells contain at least one node in the set P"; we get

$$Pr[W_i] = \frac{\binom{q}{i} i! i^{d-i}}{q^d} = \frac{q! i^{d-i}}{(q-i)! q^d}. \quad (3)$$

Since d nodes can locate in at most d different cells, we can deduce the probability that none of the nodes in Q can hear any of the nodes in P as

$$Pr\left[\overline{D}\right] = \sum_{i=1}^{d'} Pr[\epsilon \mid W_i] = \sum_{i=1}^{d'} \left(\frac{q-i}{q}\right)^{d'} \cdot Pr[W_i]. \quad (4)$$

Thus, the detection probability in one epoch, which represents the probability that at least one of the nodes in Q can hear one of the nodes in P, can be calculated as $Pr[D] = 1 - Pr[\overline{D}]$. And the detection probability in j slots can be calculated as

$$\begin{aligned} Pr\left[D_j\right] &= 1 - Pr\left[\overline{D}_j\right] \\ &= 1 - \left(\sum_{i=1}^{d'} \left(\frac{q-i}{q}\right)^{d'} \cdot Pr[W_i]\right)^j. \end{aligned} \quad (5)$$

For evaluation of the detection probability in a detection circle, we use a function to calculate how many of the regular hexagon cells are needed to cover the circle area. As shown in Figure 1, we put one regular hexagon in the center of the circle and then put a round of regular hexagons just adjacent to the inner one; we put additional rounds of regular hexagons adjacent to the inner round iteratively until the circle is fully covered.

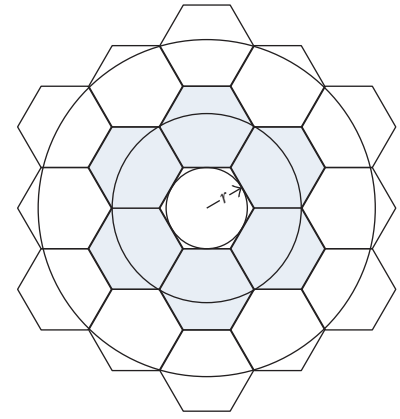

Figure 1: Covering the detection area.

For k rounds of regular hexagons, the radius X of the covered circle can be calculated as

$$X = \begin{cases} r, & k = 1, \\ 2(k-1)r, & k > 1 \text{ and } k \text{ is odd}, \\ \dfrac{(3k-2)r}{\sqrt{3}}, & k > 1 \text{ and } k \text{ is even}. \end{cases} \quad (6)$$

Therein $r = \sqrt{3}R/4$ and the number of the cells can be computed by $q = 1 + \sum_{i=1}^{k} 6(i-1) = 1 + 3k(k-1)$.

To explicitly show the detection possibility, we use some practical parameters to evaluate the detection probability. Let $R = 50$ m and $v_{\max} = 10$ m/s; the slot T is 1 s and the epoch Δ is 10 s. In a slot, the radius of the detection circle is $X_1 = ML + R = v_{\max} * \Delta + R = 3R$ and $q = 61$. In an epoch, the radius of the detection circle is $X_2 = 2\Delta + R = 2v_{\max} * \Delta + R = 5R$ and $q = 127$. We set $d' = d$ and vary d from 4 to 18 to calculate the probabilities in one slot and in five slots, respectively. We take HOP [9] as the compared scheme; suppose the deployment area is 1000 m $*$ 1000 m, and the deployment area can be covered with at least $q = 1000 * 1000/(3\sqrt{3}R^2/4) = 462$ cells.

Table 1: Detection probability in one slot.

d	4	6	8	10	12	14	16	18
HOP	7.93%	7.93%	31.06%	44.96%	58.40%	70.24%	79.87%	87.13%
Ours	64.86%	92.48%	99.20%	99.95%	100%	100%	100%	100%

Table 2: Detection probability in 5 slots.

d	4	6	8	10	12	14	16	18
HOP	33.83%	63.32%	84.42%	94.95%	98.75%	99.77%	99.97%	99.97%
Ours	79.44%	97.92%	99.93%	100%	100%	100%	100%	100%

The comparison of detection probability of one slot is shown in Table 1.

The comparison of detection probability of 5 slots is shown in Table 2.

The comparison data demonstrates that the detection rate also falls apparently, when the density of nodes declines. The reason is intuitive that the witnesses of the node and witnesses of its replica tend to have lower possibility of meeting each other due to the decrease of the witnesses while the detection area remains unchanged.

While HOP has to detect the replicas in the full deployment area, a much smaller area which is restricted by the location cert and the maximum moving speed needed to carry out the detection in our scheme. As far as a transient replica is concerned, the proposed scheme can successfully detect that in a probability higher than 90%, which is five times HOP, while the number of neighbors d is only 6.

With more detection rounds, the HOP scheme will finally find the replication from the experiments in HOP. The proposed scheme outperforms HOP in shorter detection delays.

Furthermore, the local detection area in our scheme may shrink as the maximum moving speed descends. In the scenario that nodes move trivially, the detection rate may rise as the number of cells decrease.

In the ideal case that routing protocol performed with zero packet loss rate and 100% global connectedness, global detection rate can be maintained at 1. But from this perspective of the realistic situation that nodes are often deployed in harsh environments, the consummate situation may not be achieved.

DSR [11, 13, 14] uses sources routing rather than hop-by-hop routing, with each packet to be routed carrying a complete, ordered list of nodes in its header, through which the packet must pass. The key advantage of source routing is that intermediate nodes do not need to maintain up-to-date routing information in order to route the packets they forward, since the packets themselves already contain all the routing decisions. This fact, coupled with the on-demand nature of the protocol, eliminates the need for the periodic route advertisement and neighbor detection packets present in other protocols. Based on the paper [15], Fall and Varadhan have extended the *ns*-2 [16] network simulator to accurately model the MAC and physical-layer behavior of the IEEE 802.11 wireless LAN standard, including a realistic wireless transmission channel model, and have presented the results of simulations of networks of 50 mobile nodes. Setting the pause time to 0, the performance of DSR was very good at all mobility rates and movement speeds, and the highest packet delivery ratio was obtained reaching 97.5%.

In real networks, there are other factors, time delays [16], that would affect the global detection, including the transmission delay [17] and the propagation delay [18]. We only consider the single data packet routing, a node issued a certificate, after node path forwarding until the destination, if each node has to transmit delay, and then the total number of the transmission delays of this path is the number of the path nodes. A certain distance will produce wave propagation delay in the transmission channel. In the network model we set, the maximum distance of transmission is the diagonal of the region area. Transmission delay is a function of packet's length and is not affected by the distance between two nodes. The propagation delay is defined as the time it takes to transfer a certain number of bytes over a medium. Propagation delay is the distance between the two routers divided by the propagation speed.

In order to discuss the influence of time delay on the network connectivity, we assume that the length of the packet is 1024 bytes, and the bandwidth is 10 MB/S. The maximum distance is the area of the diagonal distance of about 141 m. The total number of nodes is 1000, and the transmission rate of electromagnetic wave [19] in the air medium is slightly less than the speed of light, which approximately equals $3 * 10^8$ m/s. So the maximum transmission delay under these assumptions is 0.078 s and the maximum propagation delay is $4.7 * 10^{-7}$ s, which is considered to be negligible. Moreover, the actual values in working conditions are always much smaller than this theoretical maximum. If we set $v_{\max}$ to be 10 units/S, it will not impact on the detection process since the corresponding distance transfer 0.78 would be shorter than the communication range of a node.

Assuming that all nodes are loose time synchronization, local detection and global detection are actually performed simultaneously. If two nodes satisfy the condition $|l_k - l_j| > 2ML + R$, nodes start the local detection procedure; after the information of each node is exchanged, it is easily detected that the same ID nodes at the same time appeared in different locations. Because the area is small, the detection speed will be faster, therefore obtaining the local detection probability $Pr[D_j]$. When the distance between the capture node and the replicas is larger than $2ML + R$, at the same time, the global detection in the t_j slot of an epoch time would start acquiring the corresponding detection rate.

Table 3: Communication overhead, storage overhead, and computation overhead of detecting a replica during one slot.

Protocols	Communication	Storage	Computation
HOP	$O(d^2 f)$	$O(d^2 f + fd)$	$O(2d^2 f)$
TD	$O(d^2 f)$	$O(d^2 f)$	$O(d^3 f)$

5.2. Overhead

Communication. We evaluate the average number of location claims sent and received by a node per slot. During the local detection, the neighboring nodes exchange their location claims and the store f slots of location claims. Thus one node may send $fd+1$ location claims and receive $(fd+1)d$ location claims per slot in average. For global detection, the average number of one-hop neighbors' location certs forwarded by one node is about $2/f$ per slot. Thus a node may receive $2/f$ location certs per slot in average. In addition, if we estimate the number of nodes in the forwarding path as $\sqrt{N}$, the average number of nonneighbors' location certs forwarded by one node is about $(\sqrt{N} - 2)/f$ per slot. Then, we can derive an average overhead, per node, of $O(d^2 f + \sqrt{N}/f)$ node locations received, per slot. If N is less than f^4, then the communication overhead is less than $O(d^2 f + \sqrt{f^4}/f)$, which can be simplified to $O(d^2 f)$. For a network contains 10000 nodes, we should set $f \geq 10$ to satisfy the above condition.

Storage. A node receives altogether $(fd + 1)d$ location claims and $2/f$ location certs per slot in average. A node must allocate storage for the fd locations received in the latest f slots. Then the storage overhead amounts to $O(d^2 f)$.

Computation. In average, each sensor receives $(fd + 1)d$ location claims and $2/f$ location certs. For each of the f slots of location claims, a node should compare $d+1$ $ClaimList[t]$ with an average length d, and the computation overhead is $O(d^3)$. Thus the total overhead for the local detection is $O(d^3 f)$. In addition, a node should execute $2/f$ times of comparison for the global detection in average.

From the above analysis, we can draw a conclusion that the global detection only incurs additional overhead which is less than the overhead yielded by the local detection. The benefits introduced by the global detection have been obtained with reasonable cost. Table 3 gives the overhead comparison for the replica detection schemes. In comparison with HIP/HOP, if $f = h$ and $N \leq f^4$, the proposed scheme has an equivalent communication complexity as HIP/HOP.

The overhead of detection protocols in one slot is shown in Table 3.

6. Simulation and Discussion

We simulated the proposed mobile replica detection scheme in a mobile sensor network. Inspired by [9], we chose to use our own simulator to run fast experiments and to focus on the detection performance of the simulated scheme. In order to have a fair comparison, the duration of a time slot in the real time unit in terms of the second has not been specified and the distance is also measured with the relevant unit in the simulation experiments. A modified Random WayPoint model is used as the mobility model. Nodes are unaware of their velocities and directions but have a known maximum velocity v_{max}. Instead of choosing a certain speed for the destinations, nodes randomly vary their speed at each movement. The pause time is set to 0, so the node starts for the next destination immediately after one round of the trip. We assume that the default value of communication range R is set to 5 units and all the nodes are uniformly deployed in a $100 * 100$ square area. The default value of the maximum velocity is set to 10 units/s. The default value of the count of the nodes N is set to 1000. In this situation, the average neighborhood density in the network is 7.85.

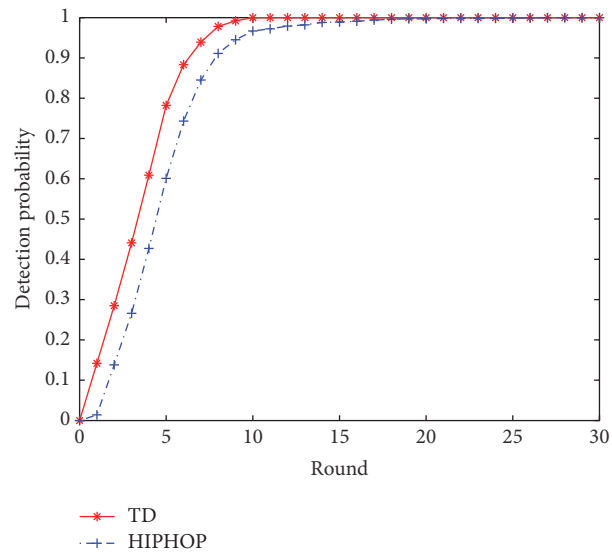

Figure 2: Detection probability when the parameter is $H = 10$, $v = 10$, $c = 1$, $R = 5$.

6.1. Detection Performance. To compare our proposals with the state of the art for replica detection in MSNs, we consider HOP as our main competitor, because of its appreciable detection performance and concise implementation.

6.1.1. Detection Performance Comparison with Default Parameters. Average degree is $H = 10$, $v = 10$, $c = 1$, $R = 5$. Setting the transmission range to $R = 5$, we simulated a network where a deployed sensor experiences an average neighborhood density of 8. Figure 2 also clearly shows that TD is faster in the clone detection compared to HIPHOP: in all the settings, HIPHOP obtain the same detection rate after 20 rounds.

6.1.2. Detection Performance Comparison When the Average Neighborhood Density Changes. If the count of nodes is fixed, the average neighborhood density varies if the communication range R changes. We performed the simulations considering $h = 10$ and setting the communication range R

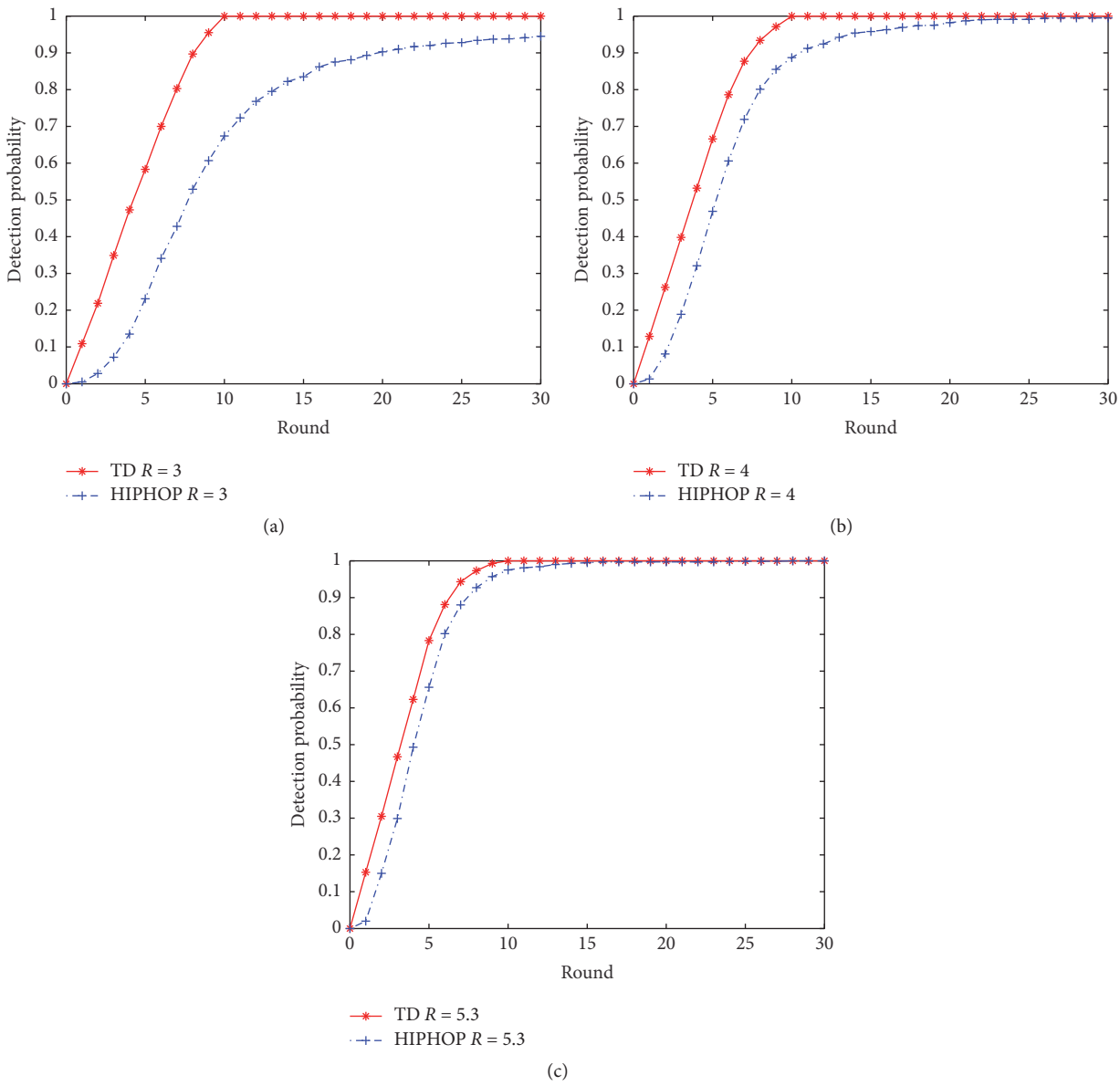

Figure 3: Detection probability for changeable parameter R.

to 3, 4, and 5.3, obtaining an average number of neighbors $d = 2$, 4, and 8, respectively.

Setting the transmission range to $R = 3$, we obtained considerable gap between TD and HOP, and all the TD settings exhibit a quite effective detection rate, since the clone is caught with a probability always reaching 1. Looking at round 5, we have that HOP detects the clone only with a probability of barely 0.2, while TD has 0.6 detection probability. When TD reaches the best detection rate (round 10), the gap with HIPHOP is about 0.30 and it is filled in around 40 (not shown in the chart) rounds.

It is obvious to notice that when the density of nodes increases (Figure 3), the detection rate of HOP and TD also goes up apparently: the reason is obviously that the clone leaves more traces in the network and the increased number of neighbors severely incurs the propagation of its location in the network. The witnesses of the node and for TD witnesses of its replica tend to have a higher possibility of meeting each other due to the increase of the witnesses while the detection area remains unchanged.

6.1.3. Detection Performance Comparison When the Length of History log h *Changes.* In the proposed scheme, we measured the time axis with a coarse unit-epoch and a fine unit-slot. However, reference scheme HOP uses "round" to measure time. Hereby, we assume that a time slot in our scheme equals a round in HOP. Thus, the count of the slots in one epoch plays the same role as the length of the history log in HOP. We change the length h with values 5, 10, 15, and 20, and the impact of h to the detection performance is shown in Figure 4.

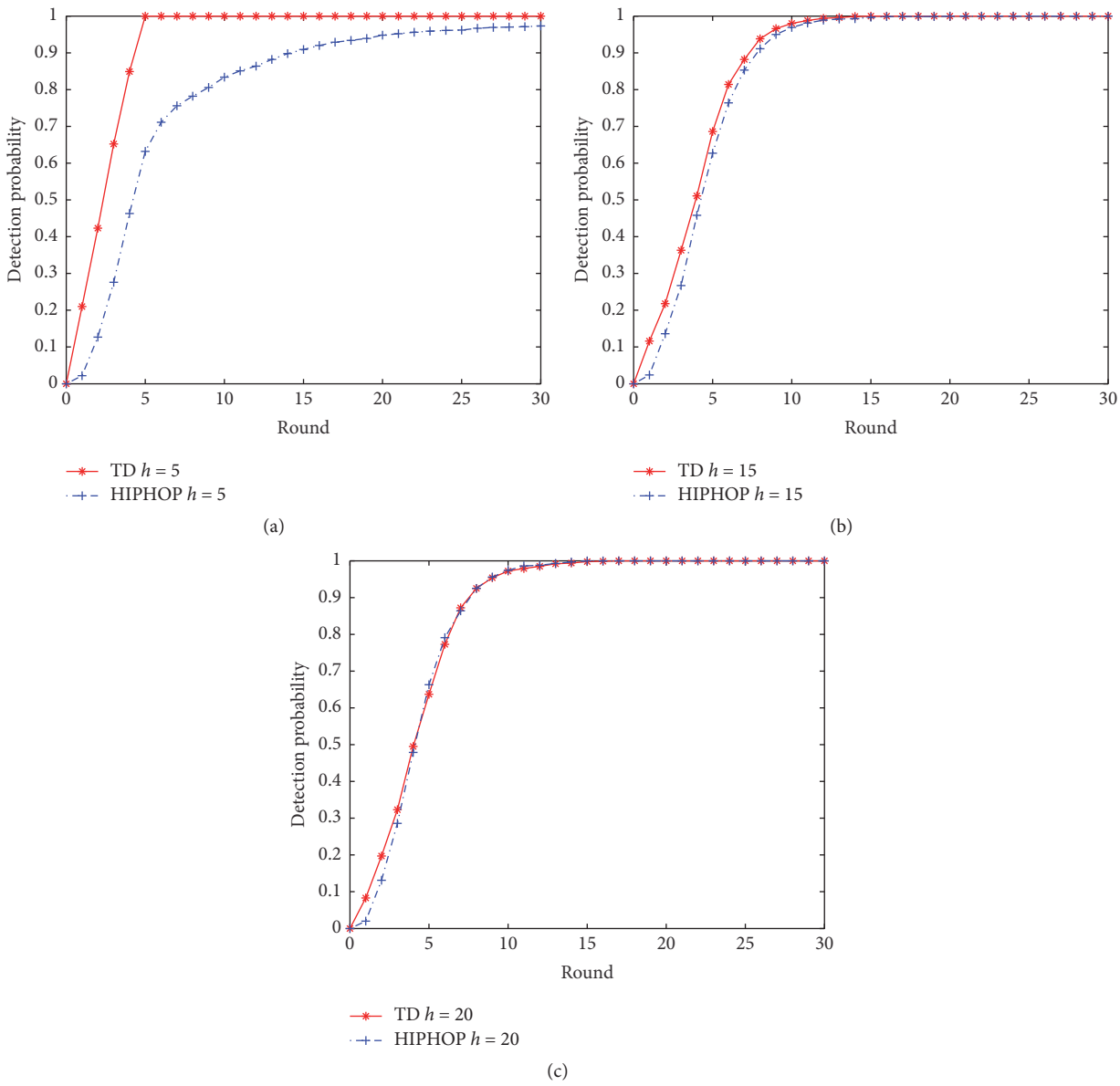

Figure 4: Detection probability for changeable parameter h.

In the setting that resulted in Figure 4, the most effective system setting is (predictably) TD with the lowest history $\log(h)$, that is, when each sensor logs $h = 5$ rounds. Hence, the cumulative detection probability curves suddenly stop. In the case of HOP, instead, adversary traces keep staying in the network (if there) and will be eventually detected, even if after many rounds. For example, to reach the same 1 detection rate, HOP needs 40 rounds (not shown in the chart). Comparing the results of the three protocols we can see as the speed of detection is very different: looking at round 4, we have that HOP detects the clone only with a probability of barely 0.45, while TD have 0.85 detection probability. When TD reach the best detection rate (round 5), the gap with HOP is about 0.4 and it is filled in around 40 (not shown in the chart) rounds.

When sequence number of the round exceeds the value of h, nodes start overwriting the history log, hence deleting the possible evidence for replica detection. Therefore, a larger h means longer preservation of the possible evidence which benefits the detection. However, storage for the history log is enlarged meanwhile.

The gap between two lines is smaller when $h = 20$: the detection rate decreases from 1 to 0.7 for TD. Again, results for HOP are opposite: things go better when h increase (there is even no difference between two protocols when h =20). As above, the constraints introduced by the RWP model slow down the information propagation and negatively impact the detection capability of our protocol.

6.1.4. Detection Performance Comparison When the Maximum Speed v_{max} Changes. We set v_{max} with values 2, 4, 6, 8, and 10. Figure 5 shows the detection probability for varying settings of the maximum moving speed of HOP

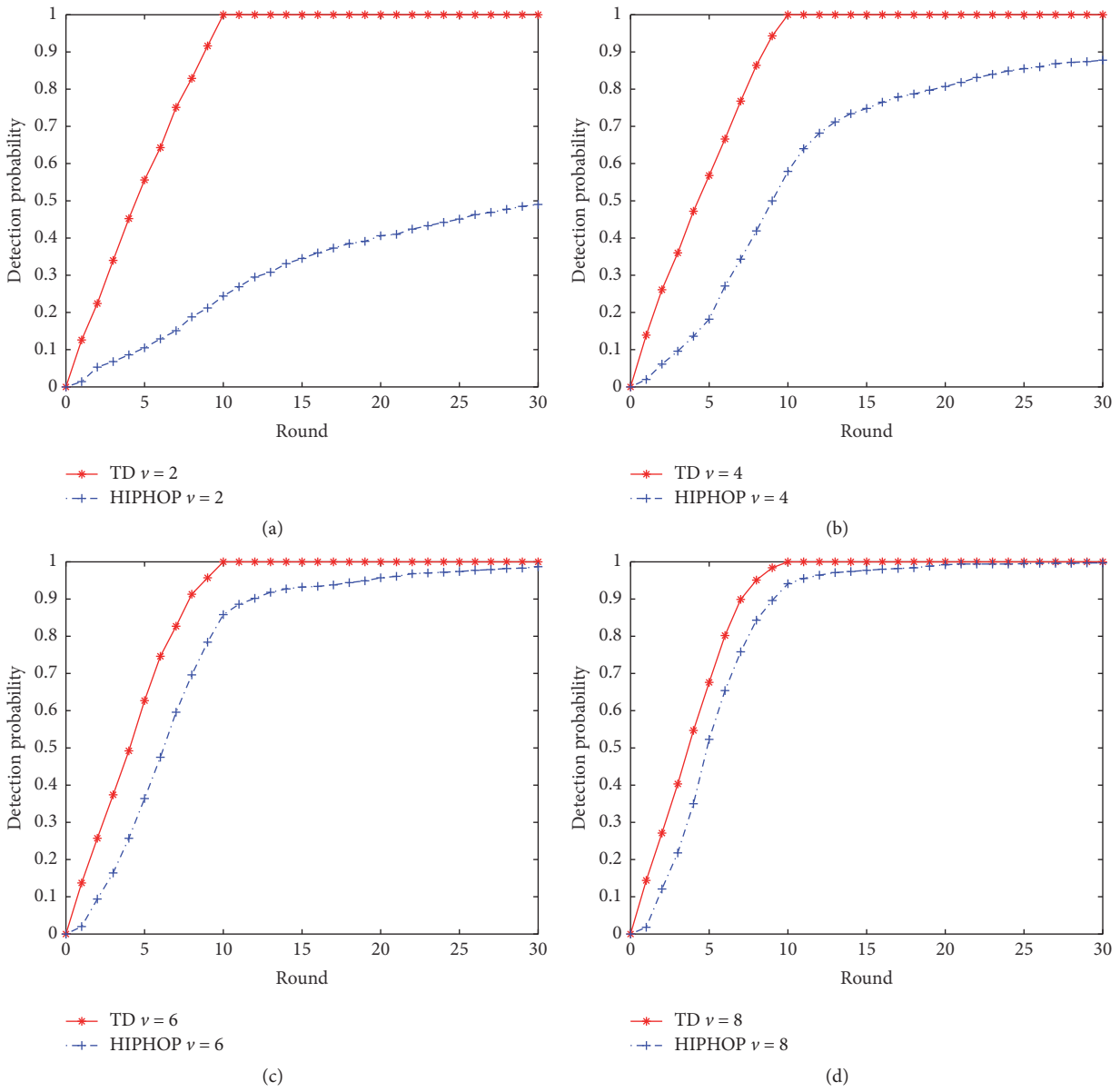

Figure 5: Detection probability for changeable parameter v.

and TD. Note that for v is 2, the detection probability is about 99% in 10 rounds for TD, but the detection probability is 0.2, which means that TD has finished detecting while HOP has just begun. With the increasing of v, the curve of detection probability for TD tends to be stable, which means regardless of the speed of nodes movement being fast and slow, our protocol can maintain high detection efficiency. The detection of HOP predominantly depends on the cooperation between nodes, and the speed of movement determines the efficiency of exchanging messages between neighbors. Nevertheless, global detection can play a bigger role when the speed is low, and the contrast of the target destination can swiftly detect the replication attack.

6.1.5. Detection Performance Comparison When Number of the Replicated Nodes c Changes. To complete the analysis of the adversary, we simulated one more adversary that places 5 and 10 clones in the network. For this setting, we compared TD with HOP. Again, experiments we performed with more than 1 clone do not differentiate much between HOP and our proposals, since the detection was performed always with probability 1 during the very first rounds for both solutions. For the sake of completeness, we report the results in Figure 6.

7. Conclusion

A novel node replica detection scheme is proposed to mitigate the threat to mobile sensor networks. In comparison with the existing localized detection scheme, the detection area of our scheme is lowered to a zone determined by the maximum distance a node can move in a time epoch. Benefiting from the decrease of the detection area, the detection probability

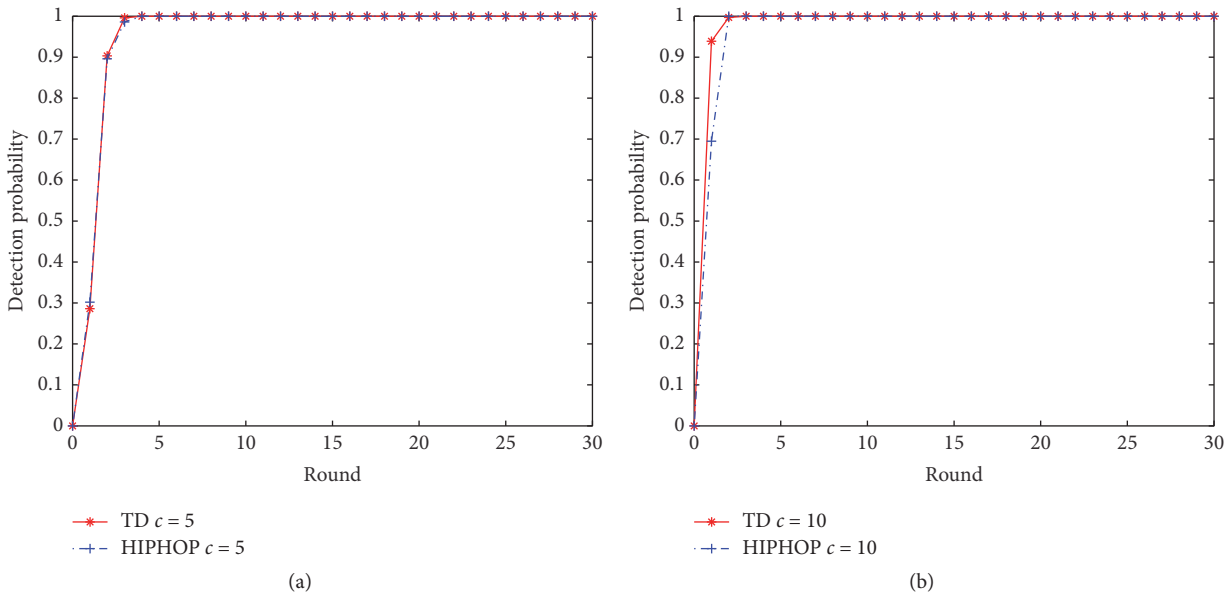

Figure 6: Detection probability for changeable parameter c.

is much higher than HOP, a recent efficient distributed detection scheme. We also demonstrated that the overhead introduced by TD is lower. The global detection step can effectively detect replicated nodes far from each other even in a network where nodes move trivially, which is more difficult for some of the existing distributed schemes. Our schemes are affected by different routing protocols; therefore they are applicable to a wide range of mobile networks. Both analysis and extensive simulations provide assistance to the quality and feasible of our scheme.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by the Natural Science Foundation of Tianjin City under Grant 11JCYBJC00800 and National Natural Science Foundation of China under Grants 60970016 and 61173032.

References

[1] B. Parno, A. Perrig, and V. Gligor, "Distributed detection of node replication attacks in sensor networks," in *Proceedings of the 26th IEEE Symposium on Security and Privacy (S and P '05)*, pp. 49–63, Oakland, CA, USA, May 2005.

[2] M. Conti, R. di Pietro, L. V. Mancini, and A. Mei, "A randomized, efficient, and distributed protocol for the detection of node replication attacks in wireless sensor networks," in *Proceeding of the 8th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '07)*, pp. 80–89, New York, NY, USA, September 2007.

[3] M. Conti, R. Di Pietro, L. Mancini, and A. Mei, "Distributed detection of clone attacks in wireless sensor networks," *IEEE Transactions on Dependable and Secure Computing*, vol. 8, no. 5, pp. 685–698, 2011.

[4] B. Zhu, S. Setia, S. Jajodia, S. Roy, and L. Wang, "Localized multicast: efficient and distributed replica detection in large-scale sensor networks," *IEEE Transactions on Mobile Computing*, vol. 9, no. 7, pp. 913–926, 2010.

[5] M. Zhang, V. Khanapure, S. Chen, and X. Xiao, "Memory efficient protocols for detecting node replication attacks in wireless sensor networks," in *Proceedings of the 17th IEEE International Conference on Network Protocols (ICNP '09)*, pp. 284–293, October 2009.

[6] J.-W. Ho, M. Wright, and S. K. Das, "Fast detection of replica node attacks in mobile sensor networks using sequential analysis," in *Proceedings of the 28th IEEE Conference on Computer Communications (IEEE INFOCOM '09)*, pp. 1773–1781, Rio de Janeiro, Brazil, April 2009.

[7] K. Xing and X. Z. Cheng, "From time domain to space domain: detecting replica attacks in mobile ad hoc networks," in *Proceedings of the IEEE (INFOCOM '10)*, San Diego, CA, USA, March 2010.

[8] C.-M. Yu, Y.-T. Tsou, C.-S. Lu, and S.-Y. Kuo, "Localized algorithms for detection of node replication attacks in mobile sensor networks," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 5, pp. 754–768, 2013.

[9] M. Conti, R. Di Pietro, and A. Spognardi, "Clone wars: distributed detection of clone attacks in mobile WSNs," *Journal of Computer and System Sciences*, vol. 80, no. 3, pp. 654–669, 2014.

[10] C. Zhou and Z. Wang, "An two dimension detection to node replication attacks in mobile sensor networks," in *Proceeding of The 10th IEEE International Conference on Anti-Counterfeiting, Security, and Identification IEEE (ASID '16)*, Xiamen, China, 2017.

[11] C. H. Foh, G. Liu, B. S. Lee, B.-C. Seet, K.-J. Wong, and C. P. Fu, "Network connectivity of one-dimensional MANETs with random waypoint movement," *IEEE Communications Letters*, vol. 9, no. 1, pp. 31–33, 2005.

[12] G. Gaubatz, J. peter Kaps, and B. Sunar, "Public key cryptography in sensor networks," *ESAS*, 18 pages, 2004.

[13] A. Liu and P. Ning, "TinyECC: a configurable library for elliptic curve cryptography in wireless sensor networks," in *Proceedings of the 7th International Conference on Information Processing in Sensor Networks (IPSN '08)*, pp. 245–256, St. Louis, Mo, USA, April 2008.

[14] D. Kukreja, S. K. Dhurandher, and B. V. R. Reddy, "Enhancing the security of dynamic source routing protocol using energy aware and distributed trust mechanism in manets," *Advances in Intelligent Systems and Computing*, vol. 321, pp. 83–95, 2015.

[15] A. Boukerche and Y. Ren, "A trust-based security system for ubiquitous and pervasive computing environments," *Computer Communications*, vol. 31, no. 18, pp. 4343–4351, 2008.

[16] K. Fall and K. Varadhan, "The VINT Project, UC Berkeley, LBL, USC/ISI, and Xerox PARC," in *Ns Notes and Documentation*, November 1997, http://www-mash.cs.berkeley.edu/ns/.

[17] X. Guo, M. R. Frater, and M. J. Ryan, "Design of a propagation-delay-tolerant MAC protocol for underwater acoustic sensor networks," *IEEE Journal of Oceanic Engineering*, vol. 34, no. 2, pp. 170–180, 2009.

[18] X. Xiong, L. Wu, and X. Chen, "Routing selection for wide-area protection based on communication reliability and time-delay requirement," *Automation of Electric Power Systems*, vol. 35, no. 3, pp. 44–48, 2011.

[19] A. A. Ibrahim and K. Sarabandi, "Simulation of long distance wave propagation in 2-d sparse random media: a statistical s-matrix approach in spectral domain," in *IEEE Transactions on Antennas Propagation*, pp. 2708–2720, 2014.