IJREAT International Journal of Research in Engineering & Advanced Technology, Volume 2, Issue 3, June-July, 2014
ISSN: 2320 – 8791 (Impact Factor: 1.479)
www.ijreat.org

Duplicate Node Detection Using Distributed Protocols (3D-NUP) in WSN

Saravanan.D[1], Jeba Moses.T[2], Arthibala.A[3]

[1,2]Assistant Professor, [3]Senior Lecturer
Dept of Information Technology,
IFET College of Engineering,

ABSTRACT:

In Wireless sensor networks (WSN), the wide range of communication is not that much secure when compared to limited area network. For providing some security purpose, the proposed system is designed. By designing two novel node clone detection techniques with different tradeoffs on network conditions and performance security is provided. The first one is based on a distributed hash table (DHT), it is a fully decentralized, key-based caching and checking system is constructed to catch cloned or duplicated nodes effectively. Our second distributed detection protocol, named randomly directed exploration, the protocol is mainly designed to provide good communication performance in dense sensor networks, by a probabilistic directed forwarding technique along with random initial direction and border determination. The simulation results uphold the protocol design and show its efficiency on communication overhead and satisfactory Detection probability.

Index Terms- DHT, clone attack, randomly directed exploration.

1. INTRODUCTION

Wireless sensor networks (WSNs) have gained a great deal of attention in the past decade due to their wide range of application areas and formidable design challenges. In general, wireless sensor networks consist of hundreds and thousands of low-cost resource-constrained, distributed sensor nodes, which usually scatter in the surveillance area randomly, working without attendance.

In this paper, we present two novel, practical node clone detection protocols with different tradeoffs on network conditions and performance. The first proposal is based on a distributed hash table (DHT) [1], by which a fully decentralized, key-based caching and checking system is constructed to catch cloned nodes. The protocol's performance on memory consumption and a critical security metric are theoretically deducted through a probability model, and the resulting equations, with necessary adjustment for real application, are supported by the simulations. In accordance with our analysis, the comprehensive simulation results show that the DHT-based protocol can detect node clone with high security level and holds strong resistance against adversary's attacks.

Our second protocol, named randomly directed exploration, is intended to provide highly efficient communication performance

with adequate detection probability for dense sensor networks. In the protocol, initially nodes send claiming messages containing a neighbour-list along with a maximum hop limit to randomly selected neighbours; then, the subsequent message transmission is regulated by a probabilistic directed technique to approximately maintain a line property through the network as well as to incur sufficient randomness for better performance on communication and resilience against adversary.

In addition, border determination mechanism is employed to further reduce communication payload. During forwarding, intermediate nodes explore claiming messages for node clone detection. By design, this protocol consumes almost minimal memory, and the simulations show that it outperforms all other detection protocols in terms of communication cost, while the detection probability is satisfactory.

2. PROPOSED SYSTEM

A. Network Model

We consider a large-scale, homogeneous sensor network consisting of resource-constrained sensor nodes. Analogous to previous distributed detection approaches; we assume that an identity-based public-key cryptography facility is available in the sensor network. Prior to deployment, each legitimate node is allocated a unique ID and a corresponding private key by a trusted third party. The public key of a node is its ID, which is the essence of an identity-based cryptosystem. Consequently, no node can lie to others about its identity. Moreover, anyone is able to verify messages signed by a node using the identity- based key. Let and denote the public and private Keys of node, respectively, and represent the signature signed by node.

We also assume that every sensor node can determine its geographic location and current relative time via a secure localization protocol and a secure time synchronization scheme, respectively.

There may or may not be a powerful base station in our modelled network, but there should exist a trusted role named initiator that is responsible for initiating a distributed detection procedure. Otherwise, an adversary can readily launch a denial-of-service (DoS) attack to the system by repeatedly mobilizing the sensor network to conduct the clone detection protocol and exhausting nodes energy.

B. General Detection Guidelines

Relying on the identity-based cryptography, secure localization and secure time synchronization used in our network model, node clone in sensor networks can be determined by the occurrence of nodes with same ID appearing on reasonably distant locations at a designated time. Specifically, at the beginning time of a round of detection that is specified by the initiator, the information regarding the ID and location of every node is claimed by its neighbours for the clone detection. In this sense, the neighbours of a node are its observers.

Subsequently, some nodes will be selected as inspectors to examine claiming messages for the purpose of clone detection. If an inspector successfully finds a clone, it becomes a witness, which will broadcast necessary evidence to inform all connected nodes revoking the cloned nodes.

While the initiator is presumably trusted, the other roles (observer, inspector, and witness) might be compromised by the adversary and behaviour maliciously.

The four roles in our protocols are summarized in Table

Roles	Trusted	Duty
Initiator	Yes	Start a round of detection
Observer	No	Claim neighbours IDs and location
Inspector	No	Buffer and check messages for location
Witness	No	Broadcast detection evidence

Table 1: Protocols Roles

C. Performance Metrics

The following metrics are used to measure a protocol's performance and evaluate its practicability.

- Detection probability and security level
- Communication cost
- Storage consumption
- Balance

3. DHT-BASED DETECTION PROTOCOL

The principle of our first distributed detection protocol is to make use of the DHT mechanism to form a decentralized caching and checking system that can effectively detect cloned nodes. Essentially, DHT enables sensor nodes to distributive construct an overlay network upon a physical sensor network and provides an efficient key-based routing within the overlay network. As a beginning of a round of DHT-based clone detection, the initiator broadcasts the action message including a random seed. Then, every observer constructs a claiming message for each neighbour node, which is referred to as an examinee of the observer and the message, and sends the message with probability independently. The introduction of the claiming probability is intended to reduce the communication overwork in case of a high-node-degree network. In the protocol, a message's DHT key that determines its routing and destination is the hash value of concatenation of the seed and the examinee ID. By means of the DHT mechanism, a claiming message will eventually be transmitted to a deterministic destination node which will cache the ID-location pair and check for node clone detection, acting as an inspector. In addition, some intermediate nodes also behave as inspectors to improve resilience against the adversary in an efficient way.

I. Distributed Hash Table

Before diving into the detection protocol, we briefly introduce DHT techniques. In principle, a distributed hash table is a decentralized distributed system that provides a key-based lookup service similar to a hash table: (key, record) pairs are stored in the DHT, and any participating node can efficiently store and retrieve records associated with specific keys. By design, DHT distributes responsibility of maintaining the mapping from keys to records among nodes in an efficient and balanced way, which allows DHT to scale to extremely large networks and be suitable to serve as a facility of distributed node clone detection. There are several different types of DHT proposals, such as CAN, Chord, and Pastry. Generally, CAN has least efficiency than others in terms of communication cost and scalability, and it is rarely employed in real systems. By contrast, Chord is widely used, and we choose Chord as a DHT implementation to demonstrate our protocol. However, our protocol can easily migrate to build upon Pastry and present similar security and performance results.

The technical core of Chord is to form a massive virtual ring in which every node is located at one point, owning a segment of the periphery. To achieve pseudo-randomness on output, a hash function is

used to map an arbitrary input into a -bit space, which can be conceived as a ring. Each node is assigned with a Chord coordinate upon joining the network. Practically for our protocol, a node's Chord point's coordinate is the hash value of the node's MAC address. All nodes divide the ring into segments by their Chord points. Likewise, the key of a record is the result of the hash function. Every node is responsible for one segment that ends at the node's Chord point, and all records whose keys fall into that segment will be transmitted to and stored in that node.

As the kernel of efficient key-based routing, every node maintains a finger table of size to facilitate a binary-tree search. Specifically, the finger table for a node with Chord coordinate contains information of nodes that are respectively responsible for holding the keys.

If two nodes are within the ring-segments distance, they are each other's predecessor and successor by the order of their coordinates, with respect to predefined. In theory, a Chord node only needs to know its direct predecessor and finger table. To improve resilience against network churn and enhance routing efficiency, every node additionally maintains a successor table, containing its successors. Typical values of are between 10 and 20.

II. Protocol Details

More importantly in our protocol, the facility of the successors table contributes to the economical selection of inspectors.

One detection round consists of three stages.

Stage 1: Initialization
Stage 2: Claiming neighbours information
Stage 3: Processing claiming messages

III. Security Discussions

- Validity of Detection
- Thwarting Framing Attack
- Protecting Witnesses
- Coping With Message-Discarding

4. PERFORMANCE ANALYSIS OF DHT-BASED PROTOCOL

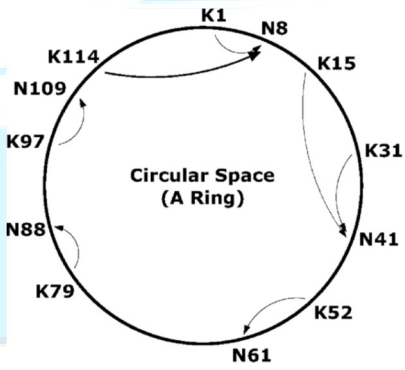
For the DHT-based detection protocol, we use the following specific measurements to evaluate its performance:

• Average number of transmitted messages
• Average size of node cache tables
• Average number of witnesses.

5. SIMULATIONS FOR DHT-BASED PROTOCOL

We implement the DHT-based detection protocol and run simulations to evaluate performance comprehensively on the OMNeT++ framework.

We design the simulations in two network scenarios. The first is an abstract network following a random graph model.



Node	Key
N8	K114
	K1
N40	K15
	K31
N61	K52
N88	K79
N109	K97

N8's Finger Table

t	Key	Node
1	72	N88
2	40	N41
3	24	N41

Fig. 1.Chord network example

By definition, a random graph is a graph that is generated by starting with a set of vertices and adding edges between them at random .The other one is a practical unit-disk graph, in which nodes are uniformly deployed in a square and follow the standard

unit-disk bidirectional communication model. In our simulations, node communication ranges are dynamically adjusted such that the average node degree approximates d.

To achieve that in a communicatively efficient way, we bring several mechanisms and effectively construct a multicast routing protocol. First, a claiming message needs to

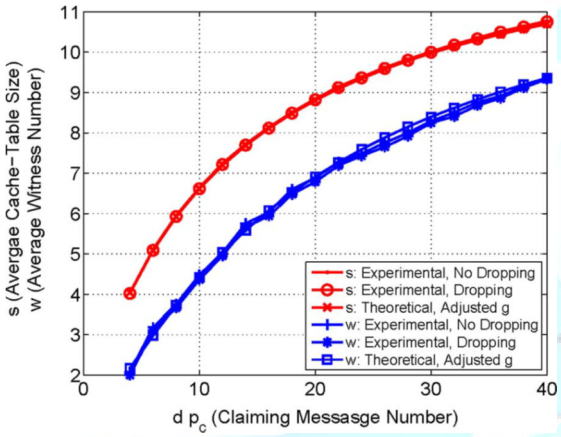


Fig.2. Simulation results for verifying performance analysis of the DHT-based detection, where is adjusted by 80% in the theoretical calculation

provide maximal hop limit, and initially it is sent to a random neighbour. Then, the message subsequent transmission will roughly maintain a line. The line transmission property helps a message go through the network as fast as possible from a locally optimal perspective. In addition, we introduce border determination mechanism to significantly reduce communication cost. We can do all of those because every node is aware of its neighbors locations, which is a basic assumption for all witness-based detection protocols but rarely utilized by other protocols.

Algorithm 1: getnextnode(M):To determine the next node
that receives the message

1: determine ideal angle, target zone, and priority zone
2: **if** no neighbors within the target zone **then**
3: **return** NIL
4: **if** no neighbors within the priority zone **then**
5: the node closest to ideal angle
6: **else**
7: nextnode<=a probabilistic node in the priority
zone, with respect to its probability proportional to
angle distance from priority zone border
8: **return** nextnode

Essentially, Algorithm 1 contains the following three mechanisms

- .Deterministic directed transmission
- Network border determination
- Probabilistic directed transmission

Analysis:

- Memory Requirement
- Communication Cost
- Security.

6. EXPERIMENTAL RESULTS FOR RANDOMLY DIRECTED EXPLORATION

We implement the randomly directed exploration protocol on the same simulation framework as the previous protocol. Since the randomly directed exploration protocol relies on a local network topology, the random graph model cannot server for its simulations. Instead, we take the unit-disk graph as the sole network scenario. We choose a constant node degree and select as the priority range of the protocol. As a result, there are an average 2.5 neighbours in the priority zone of a node. while its detection probability is satisfactory, higher than that of line-selected multicast scheme

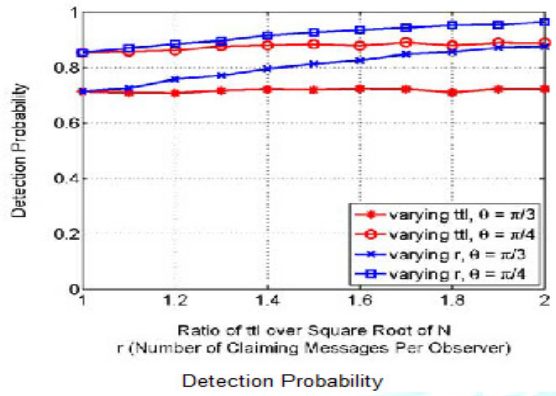


Fig.3.Performance by adjusting

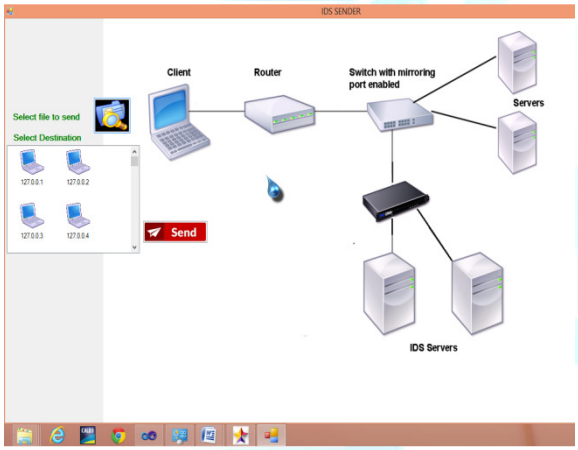


Fig.4 Sender Module

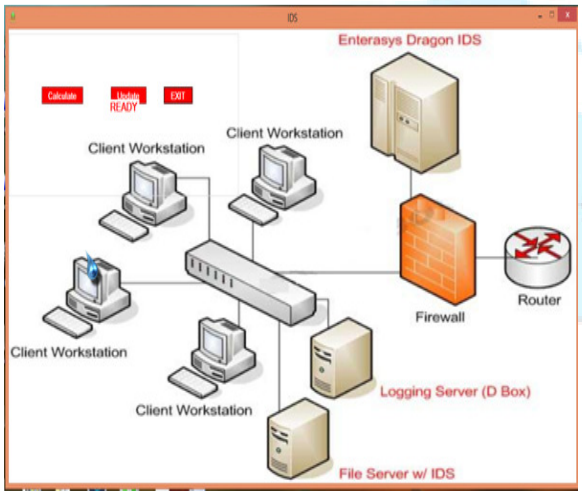


Fig.5 IDS Module

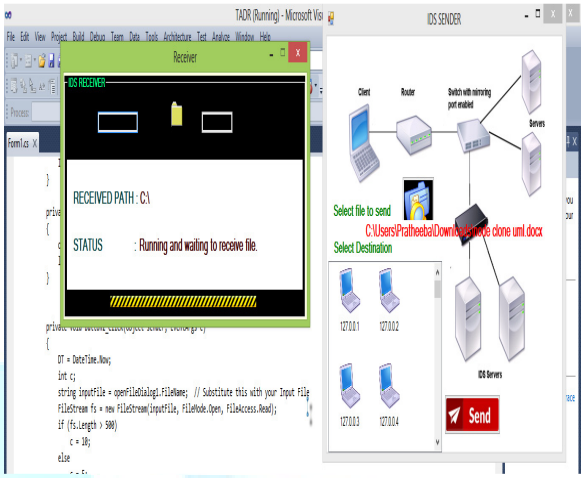


Fig.6 Result for File Transferring

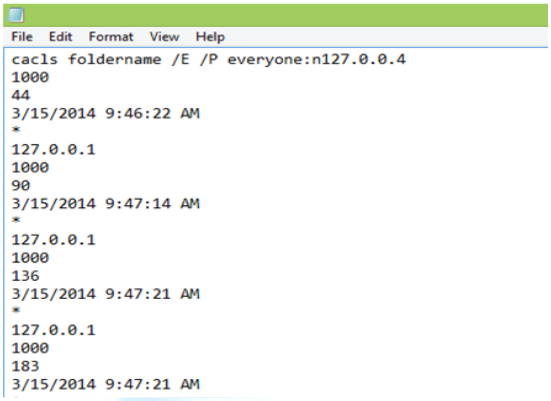


Fig.7 Checking by hash value and key value

7. CONCLUSION

Sensor nodes lack tamper-resistant hardware and are subject to the node clone attack. In this paper, we present two distributed detection protocols: One is based on a distributed hash table, which forms a Chord overlay network and provides the key-based routing, caching, and checking facilities for clone detection, and the other uses probabilistic directed technique to achieve efficient communication overhead for satisfactory detection probability. While the DHT-based protocol provides high security level for all kinds of sensor networks by one deterministic witness and additional memory-efficient, probabilistic witnesses, the randomly directed exploration

presents outstanding communication performance and minimal storage consumption for dense sensor networks.

REFERENCES

[1] B. Parno, A. Perrig, and V. Gligor, "Distributed detection node replication attacks in sensor networks," in *Proc. IEEE Symp. Security Privacy*, 2005, pp. 49–63

[2] Y. Zhang,W. Liu,W. Lou, and Y. Fang, "Location-based compromise tolerant security mechanisms for wireless sensor networks," *IEEE J.Sel. Areas Commun.*, vol. 24, no. 2, pp. 247–260, Feb. 2006.

[3]S. Zhu, S. Setia, and S. Jajodia, "LEAP: Efficient security mechanisms for large-scale distributed sensor networks," in *Proc. 10th ACM CCS*, Washington, DC, 2003, pp. 62–72.

[4]H. Choi, S. Zhu, and T. F. La Porta, "SET: Detecting node clones in sensor networks," in *Proc. 3rd SecureComm*, 2007, pp. 341–350

[5]L. Eschenauer and V. D. Gligor, "A key-management scheme for distributed sensor networks," in *Proc. 9th ACM Conf. Comput. Commun.Security*, Washington, DC, 2002, pp. 41–47

[6]I. F. Akyildiz, W. Su, Y.Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Commun. Mag.*, vol. 40, no. 8, pp.102–114, Aug. 2002.

[7] R. Anderson, H. Chan, and A. Perrig, "Key infection: Smart trust forsmart dust," in *Proc. 12th IEEE ICNP*, 2004, pp. 206–215.

[8] M. Conti, R. D. 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 *Proc. 8thACMMobiHoc*,Montreal, QC, Canada, 2007, pp. 80–89.

[9] B. Zhu, V. G. K. Addada, S. Setia, S. Jajodia, and S. Roy, "Efficient distributed detection of node replication attacks in sensor networks," in *Proc. 23rd ACSAC*, 2007, pp. 257–267.

[10] R. Brooks, P. Y. Govindaraju, M. Pirretti, N.Vijaykrishnan, and M. T. Kandemir, "On the detection of clones in sensor networks using random key predistribution," *IEEE Trans. Syst.s, Man, Cybern. C,Appl. Rev.*, vol. 37, no. 6, pp. 1246–1258, Nov. 2007.