Middle-East Journal of Scientific Research 22 (3): 444-457, 2014
ISSN 1990-9233
© IDOSI Publications, 2014
DOI: 10.5829/idosi.mejsr.2014.22.03.21909

Distance Vector Based Detection of Node Replication Attacks in Wireless Sensor Networks

[1]K.M. Anandkumar and [2]C. Jayakumar

[1]Easwari Engineering College, Anna University, Chennai, India
[2]R.M.K. Engineering College, Anna University, Chennai, India

Abstract: Wireless sensor networks have been researched extensively over the past few years. It was initially used by the military for surveillance purposes and has since expanded into industrial and civilian uses such as weather, pollution, traffic control, healthcare applications etc. Wireless sensor networks (WSNs) use small nodes called motes with constrained capabilities to sense, collect and disseminate information in many types of applications. As sensor networks become wide-spread, security issues become a central concern, especially in mission-critical tasks. When WSN are deployed in a hostile terrain environment, security becomes extremely important, as they are prone to different types of malicious attacks. Due to the poor physical protection of sensor nodes, it is generally assumed that an adversary can capture and compromise a small number of sensors in the network. In a node replication attack, an adversary can take advantage of the credentials of a compromised node to secretly introduce replicas of that node into the network. Without an effective and efficient detection mechanism, these replicas can be used to launch a variety of attacks that undermine many sensor applications and protocols. In this paper we present a distance vector based detection of node replication attacks instead of location based detection and other detection mechanisms proposed earlier for static sensor networks and also conclude that node replication attacks happen in physical and data link layer levels. The efficiency and security of our approach are also evaluated both theoretically and via simulation.

Key words: Wireless sensor networks · Security · Node replication attack detection · Node compromise · SSiD (Set Service ID)

INTRODUCTION

Sensor networks pretense distinctive security challenges because of their inherent limitations in communication and computing. The deployment nature of sensor networks makes them more vulnerable to various attacks. Sensor networks are deployed in applications where they have physical interactions with the environment, people and other objects making them more defenseless to security threats. Inherent limitations of sensor networks can be categorized as node and network limitations. The privacy and security issues in sensor networks raises rich research questions. Dense deployment of sensor networks in an unattended environment makes sensor nodes defenseless to potential attacks. Attackers can capture the sensor nodes and compromise the network to accept malicious nodes as legitimate nodes. Hardware and software improvements will address these issues to some extent but complete secure sensor networks require deployment of countermeasures such as secure key management, secure routing and light weight encryption techniques.

Figure 1 shows the system architecture of WSN with a replicated node 'A' presents in both stationary sinks 1 and 2. The time and effort needed to inject these replica nodes into the network should be much less than the effort to capture and compromise the equivalent number of original nodes. The replica nodes are controlled by the adversary, but have keying materials that allow them to seem like authorized participants in the network. Protocols for secure sensor network communication would allow replica nodes to create pair wise shared keys with other nodes and the base station, thereby enabling the nodes to encrypt, decrypt and authenticate all of their communications as if they were the original captured node. A straightforward solution to stop replica node

Corresponding Author: K.M. Anandkumar, Easwari Engineering College, Anna University, Chennai, India.

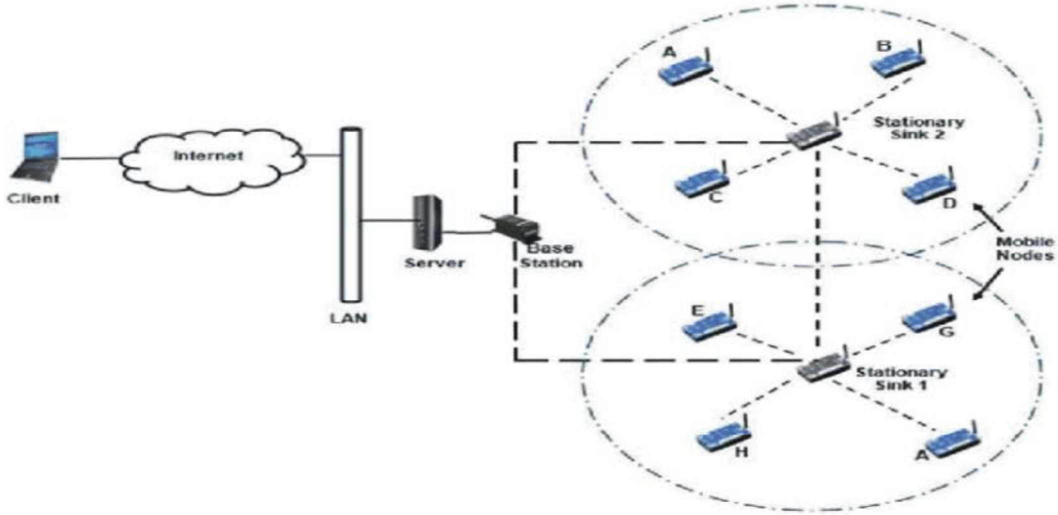


Fig. 1: System Architecture

Table 1: Attacks in wireless sensor networks

Types	Events
Sybil attacks	A single node presents multiple identities, allows to reduce the effectiveness of fault tolerant schemes such as distributed storage and multipath etc.
Spoofed, Altered	Create routing loop, attract or repel network traffic, extend or shorten source routes, generate false error messages etc.
Selective Forwarding	Either in-path or beneath path by deliberate jamming, allows to control which information is forwarded. A malicious node act like a black hole and refuses to forward every packet it receives.
Sinkhole Attacks	Attracting traffic to a specific node, e.g. to prepare selective forwarding
Wormhole Attacks	Tunneling of messages over alternative low-latency links to confuse the routing protocol, creating sinkholes etc.
Hello floods	An attacker sends or replays a routing protocols hello packets with more energy

Table 2: Layering approaches in sensor network

Layers	Attack Types
Application Layer	Subversion and Malicious Nodes
Network Layer	Wormholes, Sinkholes, Sybil, Routing loops
Data link Layer	SSiD Copying, Jamming
Physical Layer	Node capture and replication attacks

attacks is to prevent the adversary from extracting secret key materials from nodes by equipping them with tamper-resistant hardware. Table 1, shows the different kinds of attacks in WSN and Table 2, shows the layer level approach in which the node replication attacks happen in the physical layer.

In this paper we propose a distance vector based detection of node replication attacks between nodes (MOTES) and base station, instead of location based approach proposed earlier in which distance can be computed between node and base station and also between inter sensor nodes. These node replication attacks happen in physical and data link layer levels in which physically capturing the node takes place in layer 1 and copying SSiD of one node to another node thereby making the replication possible takes place in layer 2. Node capturing and replication of the nodes happen in the above said layers and clearly this process does not extend beyond these layers. So there is no point in discussing MAC and IP of data link and network layers[1].

The rest of the paper is organized as follows: Section 2 describes the related works, Section 3 presents a network assumption of our scheme, Section 4 presents the proposed distance vector based detection of replication attacks mechanism with security and performance analysis, Section 5 presents the extensive simulation results that we conducted to evaluate the proposed scheme, Section 6 presents the comparison of our method with existing methods and Finally, Section 7 concludes the paper.

Related Works: The following section describes the various mechanisms proposed so far in order to solve the node replication attacks in WSN.

One of the first solutions for the detection of clone attacks relies on a centralized Base Station (BS) [2]. In this solution, each node sends a list of its neighbors and their locations (that is, the geographical coordinates of each node) to a BS. The same node ID in two lists with inconsistent locations will result in clone detection. Then, the BS revokes the clones. This solution has several drawbacks, such as the presence of a single point of failure (the BS) and high communication cost due to the large number of messages. Further, nodes close to the BS will be required to route much more messages than other nodes, hence shortening their operational life.

Another centralized clone detection protocol has been recently proposed in [3]. This solution assumes that a random key pre distribution security scheme is implemented in the sensor network. That is, each node is assigned a set of k symmetric keys, randomly selected from a larger pool of keys [4]. For the detection, each node constructs a counting Bloom filter from the keys it uses for communication. Then, each node sends its own filter to the BS. From all the reports, the BS counts the number of times each key is used in the network. The keys used too often (above a threshold) are considered cloned and a corresponding revocation procedure is raised.

Parno *et al*. proposed the work to address the node replication attacks [5]. They proposed two protocols: Randomized Multicast and Line-Selected Multicast. In Randomized Multicast, each node broadcasts a location claim to its neighbors. Then each neighbor selects some random locations within the network and forwards the location claim with a probability to the nodes (refer to as witness nodes) closest to chosen locations by using geographic routing. According to Birthday Paradox [6], at least one witness node is likely to receive conflicting location claims when replicated nodes exist in the network. In order to reduce the communication costs and increase the probability of detection, they proposed Line-Selected Multicast protocol. Besides storing location claims in randomly selected witness nodes, the intermediate nodes for forwarding location claims can also be witness nodes. This seems like randomly drawing a line across the network and the intersection of two lines becomes the evidence node of receiving conflicting location claims.

Zhu *et al*. proposed two more efficient distributed protocols for detecting node replication attacks: Single Deterministic Cell (SDC) and Parallel Multiple Probabilistic Cells (P-MPC) [7]. Both protocols need the sensor network to be a geographic grid, each unit of which is called a cell. In SDC each node's ID is uniquely mapped to one of the cells in the grid. When executing detection procedure, each node broadcasts a location claim to its neighbors. Then each neighbor forwards the location claim with a probability to a unique cell by executing a geographic hash function [8] with the input of node's ID. Once any node in the destination cell receives the location claim, it floods the location claim to the entire cell. Each node in the destination cell stores the location claim with a probability. Therefore, the clone nodes will be detected with a certain probability since the location claims of clone nodes will be forwarded to the same cell. The difference between SDC and P-MPC is the number of destination cells. In P -MPC the location claim is forwarded to multiple deterministic cells with various probabilities by executing a geographic hash function with the input of node's ID. The rest of the procedure is similar to SDC. Therefore, the clone nodes will be detected with a certain probability as wel l[7].

Choi *et al*. proposed a clone detection approach in sensor networks called SET. In SET the network is randomly divided into exclusive subsets. Each of the subset has a subset leader and members are one-hop away from their subset leader. Next, multiple roots are randomly decided to construct multiple sub-trees and each subset is a node of the sub-tree. Each subset leader collects member information and forwards to the root of the sub-tree [8]. The intersection operation is performed on each sub-tree root to detect replicated nodes. If the intersection of all subsets of a sub-tree is empty, there are no clone nodes in this sub-tree. In the final stage, each root forwards its report to the BS. The BS detects the clone nodes by computing the intersection of any two received sub-trees. In summary, SET detects clone nodes by sending node's information to the BS from subset leader to the root node of a randomly constructed sub-tree and then to the BS [9].

Bekara and Laurent-Maknavicious proposed a new protocol for securing WSN against node replication attacks by limiting the order of deployment [10]. Their scheme requires sensors to be deployed progressively in successive generations. Each node belongs to a unique generation. In their scheme, only newly deployed nodes are able to establish pair-wise keys with their neighbors and all nodes in the network know the number of the highest deployed generation. Therefore, the clone nodes will fail to establish pair-wise keys with their neighbors since the clone nodes belong to an old deployed generation.

The only approach that achieves real-time detection of clone attacks in WSN was proposed by Xing *et al*. [11]. In their approach, each sensor computes a fingerprint by incorporating the neighborhood information through a superimposed *s*-disjunct code [12]. Each node stores the fingerprint of all neighbors. Whenever a node sends a message, the fingerprint should be included in the message and thus neighbors can verify the fingerprint. The messages sent by clone nodes deployed in other locations will be detected and dropped since the fingerprint does not belong to the same "community".

Conti *et al*. proposed a recent work for detection of node clone attacks in WSNs called RED based distributed detection [13]. When executing RED, the BS broadcasts a random value to all nodes in the network. The subsequent operations are similar to Parno *et al*.'s scheme except for the selection of witness nodes. In RED the witness nodes are selected based on a pseudo random function with the inputs of node's ID, random value which is broadcasted by the BS and the number of destination locations. Location claims with the same node ID will be forwarded to the same witness nodes in each detection phase. Hence the replicated nodes will be detected in each detection phase. When next time the RED executes, the witness nodes will be different since the random value which is broadcasted by the BS is changed.

Anandkumar et al. proposed a very recent work [14] for the detection of node replication attacks by improving the existing RED algorithm by considering the witness of witness nodes to avoid duplication in sensor networks and self evaluation schemes of base station to verify the previous history of node's location from its own memory. This improves the considerable overhead of communication cost in terms of memory and computational intelligence.

Network Assumptions: In this section describes the network assumptions that we consider and taken for our experiments. We consider static network with few original nodes and replicated node along with coordinator or base station. Also described the mode of node capturing by following two ways

- Physical node capturing
- Logical node capturing (Sniffing)

Static WIRELESS Sensor Network
Mode of Node Capture - Physically:

- Nodes are stationary / static, No movements, Fixed location
- Location of the node is constant and referred by X and Y Coordinates
- No replication or attack is possible at the of time of initialization of the network, All are trusted nodes
- After some time any one of the node from the network can be PHYSICALLY CAPTURED AND MAKE THE SIMILLAR NODE (CLONE NODE)
- Cloned node and Original node are introduced simultaneously into the network when,
- Network get restarts due to any reasons (Network resumes back after any failure – Battery depletion / drain, shut down the network for any Maintenance reasons and get back, Updating any processes (or) replacement of any node due to physical damage to any one of the node (or) some (or) all nodes
- Attacker / Intruder voluntarily introduce the network jamming by DoS attacks for some time period and in this mean time they introduced the replicated node in to the network OR
- By directly introduce one / more clone nodes while the network is running (by Force – push the node inside) – HIGHLY IMMPOSIBLE
- But network get resumes back definitely all the node are verified/checked by either physically (or) logically
- The above can be done by counting the no. of nodes in the network
- But it is difficult (some times not possible) for network containing very large no. of sensor nodes (for example more than 1000) and sparsely located like in the environmental monitoring systems
- When the initial count is 'n' and after some time it may be 'n+x' by count definitely we assume that there may by some nodes are added into the network with right authentication information's (Public and Private keys) but at the same time replication happened in the network
- n – No. of nodes at initial count
- x – Additional no. of node/nodes added into the network

Mode of Node Capture – Not Physically – by Sniffer (Reveal All the Information):

- Monitor the network traffic continuously and access all
- By capturing the packet – reveal all the information including MAC and make the similar node as a clone and introduced in the network.

In the above Fig. 2 totally seven nodes are considered. Among the nodes Node N_1, N2, N3, N4 and N_5 are original or reputed nodes and nodes N1 $_1$and N1 $_2$ are adversary nodes and try to contact directly and indirectly through some other node in multihop way.

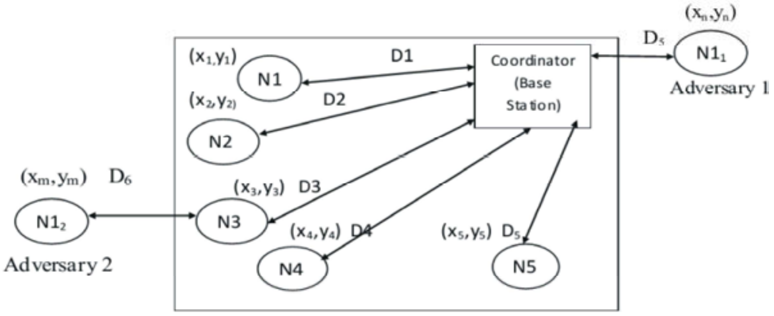


Fig. 2: Network Assumption

Finally the adversaries or called malicious nodes make data redundant at coordinator. D_1 to D_6 are distance between the nodes and the coordinator. (x_1,y_1) to (x_5,y_6) and (x_n,y_n) (x_m,y_m) are coordinators on the Cartesian plane. Finally the distance can be computed between the nodes and the coordinator using haversine formula. The above assumption of the network was simulated in mote view environment and the extensive study was conducted in the simulation section.

Distancevectorbaseddetection of Node Replication Mechanism: Distance vector based detection of node replication scheme provides the solution to detect and revocate the clone node from the network. All the above mechanisms are based on the comparisons of node's location and witness factor which needs high computational cost and time. Our method provides the simplest solution for detecting and eliminating the replicated node from the network. Consider the following network which consists of an original node, a replication node and a coordinator.

The distance between two nodes can be computed using the following formula,

$$D = \sqrt{(X_2 - X_1)^2 + (Y_2 - Y_1)^2}$$

where,
(X_1, Y_1) and (X_2, Y_2) are any two points on the Cartesian plane
Similarly the midpoint on the same plane, 'm' will be
Midpoint, $m = ((x_1 + x_2) / 2, (y_1 + y_2) / 2)$

If the sensor mote or device is equipped with GPS enabled service then the distance can be computed by Haversine formula with respect to degree, minute and seconds on the map in the following way,

Haversin ($\frac{d}{r}$) = haversin (φ_2) + cos (φ_1) cos (φ_1) haversin (ϑ_2-ϑ_1)

where,
'd' - Distance between two points
'r' - Radius of the sphere (Earth radius=6371 km)
φ_1, φ_2 - Latitude of point 1 and latitude of point 2
ϑ_2-ϑ_1 - Longitude of point 1 and longitude of point 2

One can then solve'd' either by simply applying the inverse haversine (if available) or by using the arcsine (inverse sine) function
$d = r.\text{haversin}^{-1}(h)$

$d = 2.r.\arcsin^{-1} (h)$

$D = 2.r.\arcsin \left(\sqrt{haversin(\varphi_2 - \varphi_1) + \cos(\varphi_1)\cos(\varphi_2) - \sqrt{haversin(\varphi_2 - \varphi_1)}}\right)$

$D = 2.r.\arcsin \left(\sqrt{(sin^2((\varphi_2 - \varphi_1)/2) + \cos(\varphi_1)\cos(\varphi_2) - sin^2(\varphi_2 - \varphi_1)/2)}\right)$

Fig. 3 shows the calculation of distance between the node N1 with all other nodes N2, N3, N4 and N5 present in the network.

Fig. 4 shows the calculation of distance between the adversary node 1, $N1_1$ with all other nodes N2, N3, N4 and N5 present in the network

Fig. 5 shows the calculation of distance between the adversary node 2, $N1_2$ with all other nodes N2, N3, N4 and N5 present in the network.

The Distance based method proved that it overcomes the disadvantages available in location based method. Because in location based method of identifying the replication nodes is not suitable if the location of the both trusted and replicated nodes got overlapped with the same location. Our method of identification of a replicated node completely depends on its distance from the coordinator. So, the coordinator computes the location of nodes frequently and then compares these values with the initial deployment values and removes the duplicated node from the network. If the duplicated node is too smart

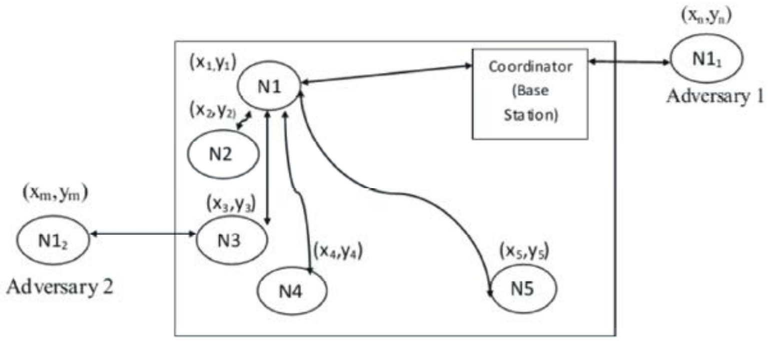


Fig. 3: Network Assumption – Distance of node N1 with other sensors in the network

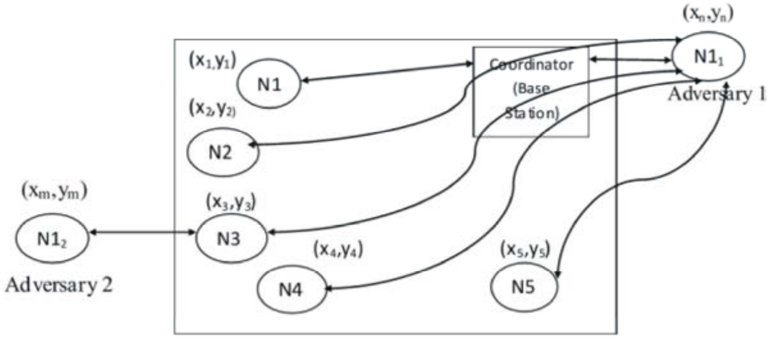


Fig. 4: Network Assumption – Distance between adversary node (1) $N1_1$ with other sensors in the network

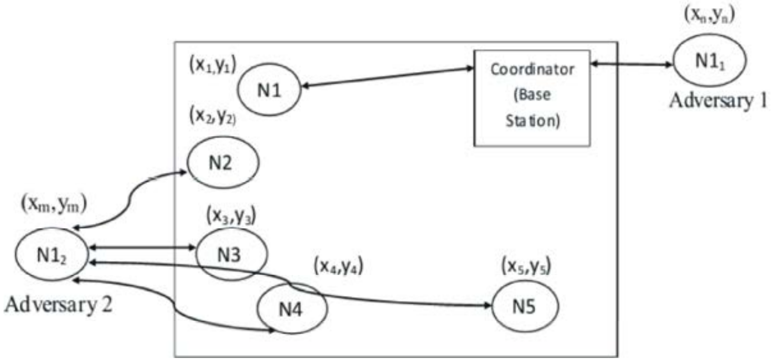


Fig. 5: Network Assumption – Distance between adversary node (2) $N1_2$ with other sensors in the network

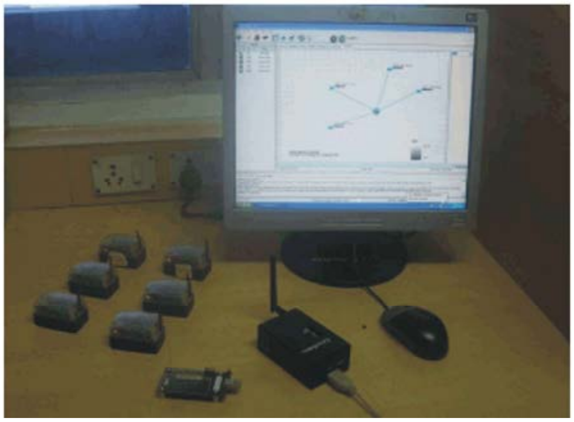
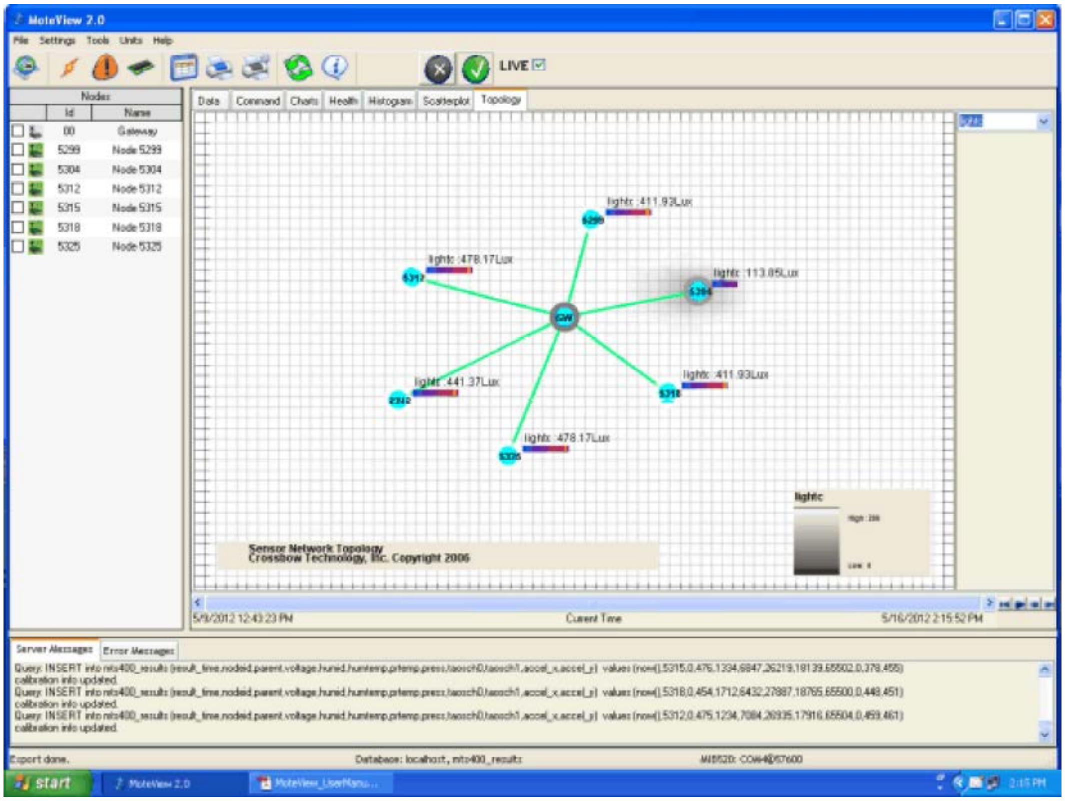


Fig. 6: Practical setup of environmental monitoring sensors with coordinator



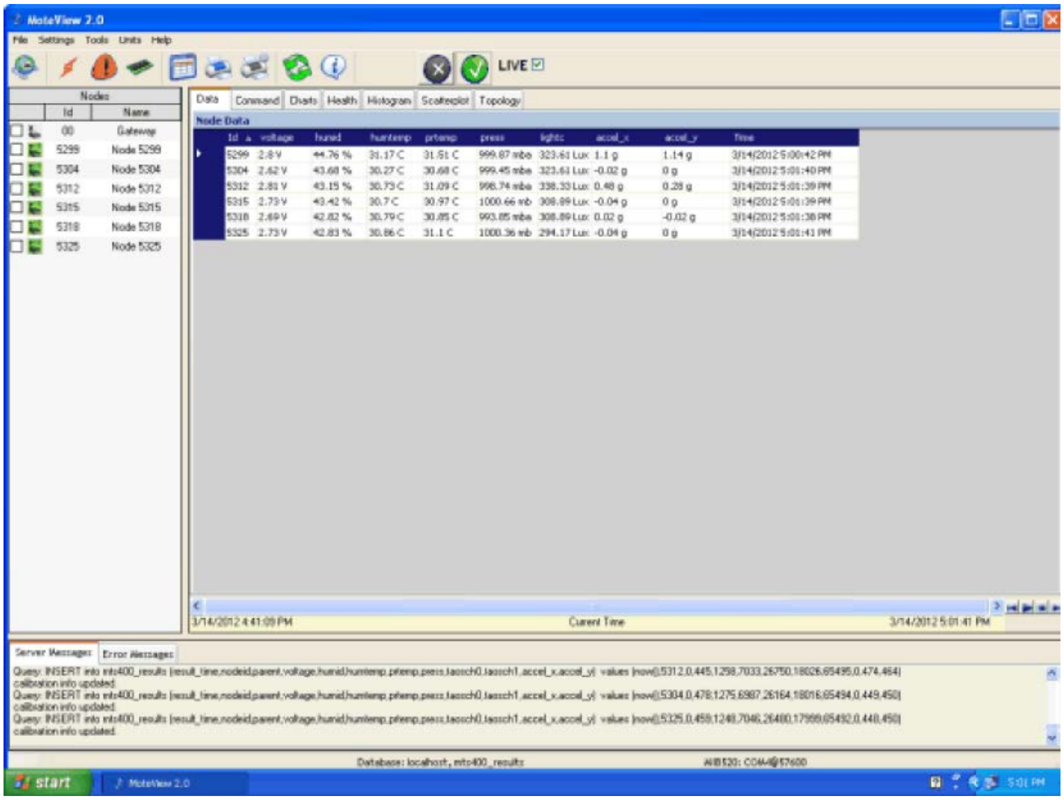
Id	Name
00	Gateway
5299	Node 5299
5304	Node 5304
5312	Node 5312
5315	Node 5315
5318	Node 5318
5325	Node 5325

5/9/2012 12:43:23 PM Current Time 5/16/2012 2:15:52 PM

```
Query: INSERT into mts400_results (result_time,nodeid,parent,voltage,humid,humtemp,prtemp,press,taosch0,taosch1,accel_x,accel_y)  values (now(),5315,0,476,1334,6847,26219,18139,65502,0,379,455)
calibration info updated.
Query: INSERT into mts400_results (result_time,nodeid,parent,voltage,humid,humtemp,prtemp,press,taosch0,taosch1,accel_x,accel_y)  values (now(),5318,0,454,1712,6432,27887,18765,65500,0,448,451)
calibration info updated.
Query: INSERT into mts400_results (result_time,nodeid,parent,voltage,humid,humtemp,prtemp,press,taosch0,taosch1,accel_x,accel_y)  values (now(),5312,0,475,1234,7084,26935,17916,65504,0,459,461)
calibration info updated.
```

Export done. Database: localhost, mts400_results MIB520: COM4@57600

Fig. 7: Topology of sensors arranged with coordinator (Gateway)


MoteView 2.0
File Settings Tools Units Help
LIVE ☑
Nodes
Data Command Charts Health Histogram Scatterplot Topology
Server Messages
Error Messages
start
MoteView 2.0
5:01 PM

Id	Name
00	Gateway
5299	Node 5299
5304	Node 5304
5312	Node 5312
5315	Node 5315
5318	Node 5318
5325	Node 5325

Node Data

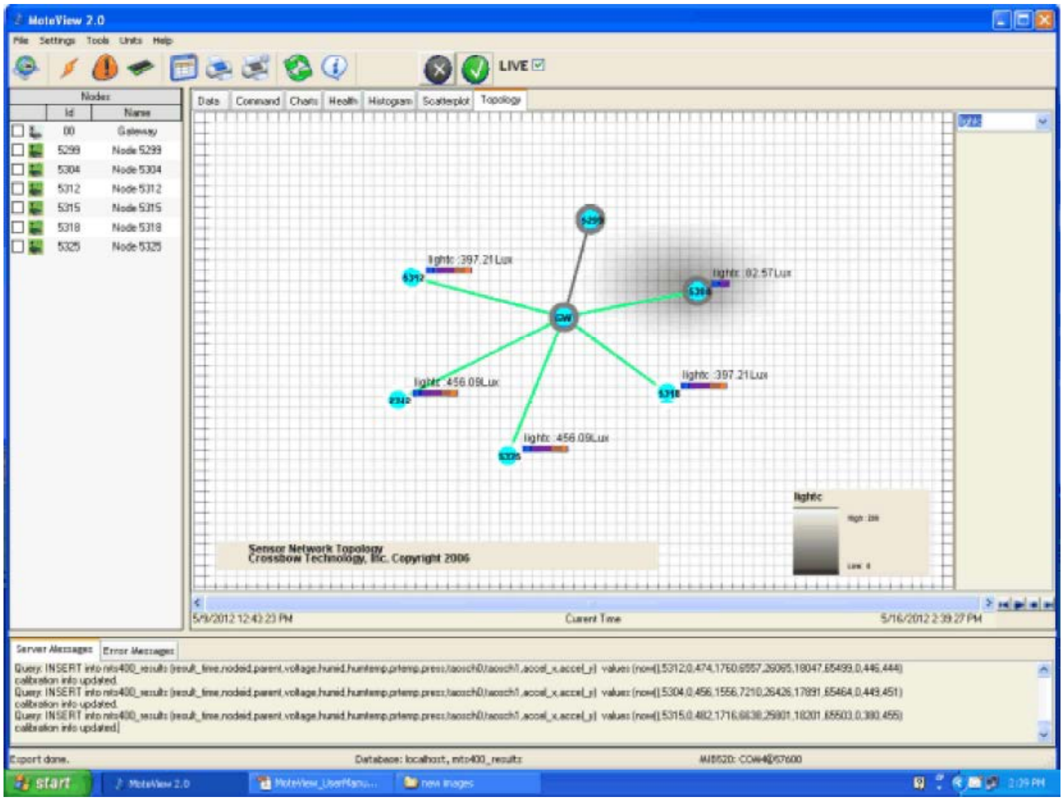
Id	voltage	humid	humtemp	prtemp	press	lightc	accel_x	accel_y	Time
5299	2.8 V	44.76 %	31.17 C	31.51 C	999.87 mba	323.61 Lux	1.1 g	1.14 g	3/14/2012 5:00:42 PM
5304	2.62 V	43.68 %	30.27 C	30.68 C	999.45 mba	323.61 Lux	-0.02 g	0 g	3/14/2012 5:01:40 PM
5312	2.81 V	43.15 %	30.73 C	31.09 C	998.74 mba	338.33 Lux	0.48 g	0.28 g	3/14/2012 5:01:39 PM
5315	2.73 V	43.42 %	30.7 C	30.97 C	1000.66 mb	308.89 Lux	-0.04 g	0 g	3/14/2012 5:01:39 PM
5318	2.69 V	42.82 %	30.79 C	30.85 C	993.85 mba	308.89 Lux	0.02 g	-0.02 g	3/14/2012 5:01:38 PM
5325	2.73 V	42.83 %	30.86 C	31.1 C	1000.36 mb	294.17 Lux	-0.04 g	0 g	3/14/2012 5:01:41 PM

3/14/2012 4:41:09 PM Current Time 3/14/2012 5:01:41 PM

```
Query: INSERT into mts400_results (result_time,nodeid,parent,voltage,humid,humtemp,prtemp,press,taosch0,taosch1,accel_x,accel_y)  values (now(),5312,0,445,1258,7033,26750,18026,65495,0,474,464)
calibration info updated.
Query: INSERT into mts400_results (result_time,nodeid,parent,voltage,humid,humtemp,prtemp,press,taosch0,taosch1,accel_x,accel_y)  values (now(),5304,0,478,1275,6987,26164,18016,65494,0,449,450)
calibration info updated.
Query: INSERT into mts400_results (result_time,nodeid,parent,voltage,humid,humtemp,prtemp,press,taosch0,taosch1,accel_x,accel_y)  values (now(),5325,0,459,1248,7046,26480,17999,65492,0,448,450)
calibration info updated.
```

Database: localhost, mts400_results MIB520: COM4@57600

Fig. 8: Recordings of the all the six sensor values



	Nodes	
	Id	Name
□	00	Gateway
□	5299	Node 5299
□	5304	Node 5304
□	5312	Node 5312
□	5315	Node 5315
□	5318	Node 5318
□	5325	Node 5325

```
Query: INSERT into mts400_results (result_time,nodeid,parent,voltage,humid,humtemp,prtemp,press,taosch0,taosch1,accel_x,accel_y) values (now(),5312,0,474,1760,6557,26065,18047,65499,0,446,444)
calibration info updated.
Query: INSERT into mts400_results (result_time,nodeid,parent,voltage,humid,humtemp,prtemp,press,taosch0,taosch1,accel_x,accel_y) values (now(),5304,0,456,1556,7210,26426,17891,65464,0,449,451)
calibration info updated.
Query: INSERT into mts400_results (result_time,nodeid,parent,voltage,humid,humtemp,prtemp,press,taosch0,taosch1,accel_x,accel_y) values (now(),5315,0,482,1716,6838,25801,18201,65503,0,380,455)
calibration info updated.
```

Export done. Database: localhost, mts400_results MIB520: COM4@57600

Fig. 9: Topology represents that node ID 5299 has removed (captured) from the network

and intelligible and it matches the same distance with the trusted node, then we compute the distance of the trusted node from all the other nodes available in the network and compare with the same evaluation as for the duplicated node. Definitely the above computed results differentiate the trusted node with duplicated or replicated nodes.

Simulations: We simulated the above environment using environmental monitoring sensors (MICA MIB 520) in mote view software. Also we simulated the same in NS-2 environment and the results were taken. Fig.6 shows six environmental sensor motes connected with the coordinator and the other end of the coordinator is connected to the simulation window to visualize the result.

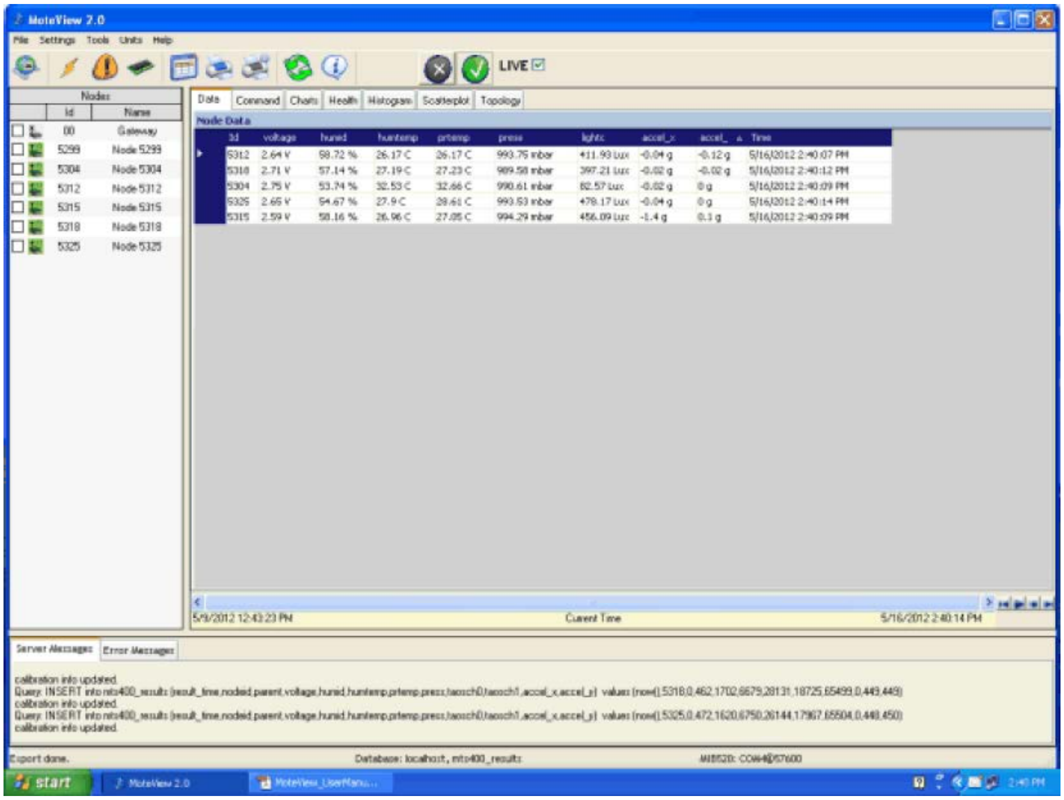
Fig. 7 shows the topology of sensors connected with the coordinator. A Central hub (GW-Gateway) acts as the coordinator and is surrounded by sensors shown with their unique Id (SSiD) under a common network group ID.

Fig. 8 shows the data recordings from the sensors for the parameters Temperature, Humidity, Light, Acceleration and Atmospheric Pressure corresponding to their IDs.

Fig. 9 of network topology shows that the sensor ID 5299 was removed or captured physically from the network and it is idle (Appears gray line). This particular sensor, subject to replication and actual ID 5299 was converted into ID 5325 which is already available in the network. After this replication, the network consists of two sensor motes with same SSiD 5325. But the replication motes have overlapped locations and they appear in a single place. Even though the network consists of 6 sensor motes of the same kind, the topology shows only 5 motes in an active state due to the above said reasons.

Fig. 10 shows the recordings of 6 sensor motes. But due to the replication of ID 5325 (Instead of 5299) recordings appear in 5 rows. So in the row of ID 5325 the values were continuously changed irrespective of timing intervals of predefined values of the sensors. Because the original or trusted node ID 5325 records its values and at the same time the replicated ID 5325 (from 5299) also records and updates the values in the same row using the time interval gap of trusted ID 5325.

Similarly Fig. 11 shows that two nodes are captured from the network and both are replicated with another ID which is already available in the network.


MoteView 2.0
File Settings Tools Units Help
LIVE ☑
Data Command Charts Health Histogram Scatterplot Topology
Nodes
start MoteView 2.0 MoteView_UserManu... 2:40 PM

Id	Name
00	Gateway
5299	Node 5299
5304	Node 5304
5312	Node 5312
5315	Node 5315
5318	Node 5318
5325	Node 5325

Node Data

Id	voltage	humid	humtemp	prtemp	press	lightx	accel_x	accel_	Time
5312	2.64 V	58.72 %	26.17 C	26.17 C	993.75 mbar	411.93 Lux	-0.04 g	-0.12 g	5/16/2012 2:40:07 PM
5318	2.71 V	57.14 %	27.19 C	27.23 C	989.58 mbar	397.21 Lux	-0.02 g	-0.02 g	5/16/2012 2:40:12 PM
5304	2.75 V	53.74 %	32.53 C	32.66 C	990.61 mbar	82.57 Lux	-0.02 g	0 g	5/16/2012 2:40:09 PM
5325	2.65 V	54.67 %	27.9 C	28.61 C	993.53 mbar	478.17 Lux	-0.04 g	0 g	5/16/2012 2:40:14 PM
5315	2.59 V	58.16 %	26.96 C	27.05 C	994.29 mbar	456.09 Lux	-1.4 g	0.1 g	5/16/2012 2:40:09 PM

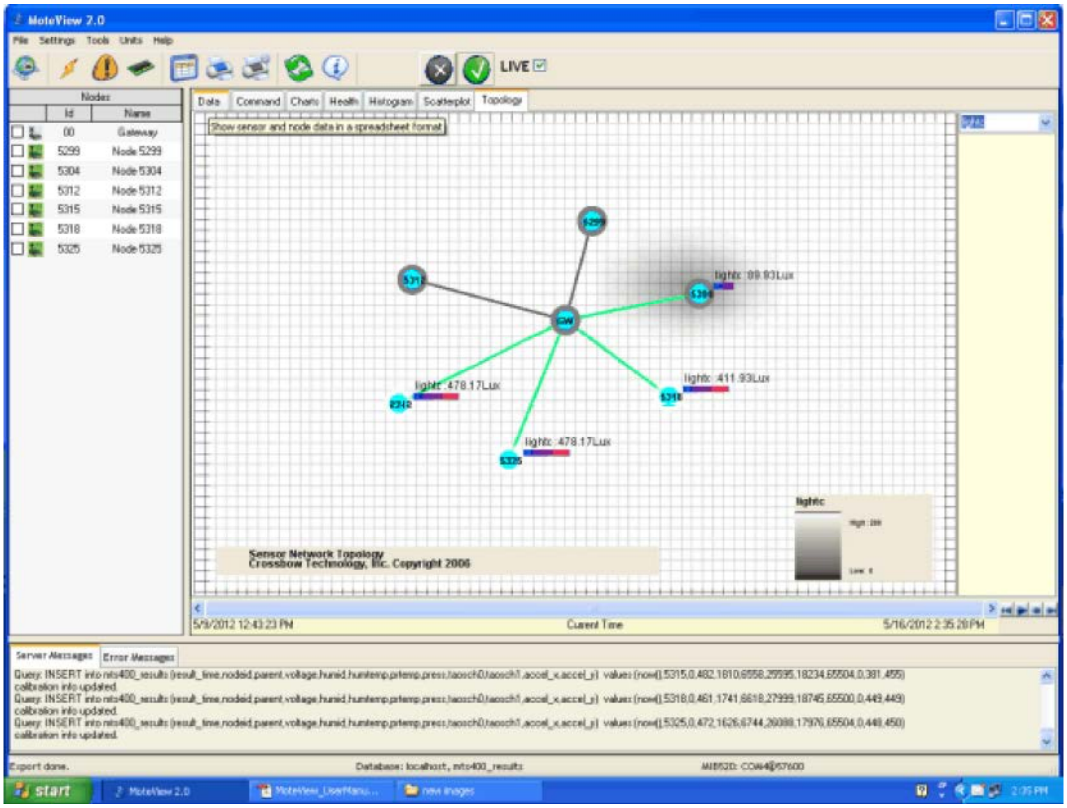
5/9/2012 12:43:23 PM Current Time 5/16/2012 2:40:14 PM

Server Messages | Error Messages

```
calibration info updated.
Query: INSERT into mts400_results (result_time,nodeid,parent,voltage,humid,humtemp,prtemp,press,taosch0,taosch1,accel_x,accel_y)  values (now(),5318,0,462,1702,6679,28131,18725,65499,0,449,449)
calibration info updated.
Query: INSERT into mts400_results (result_time,nodeid,parent,voltage,humid,humtemp,prtemp,press,taosch0,taosch1,accel_x,accel_y)  values (now(),5325,0,472,1620,6750,26144,17967,65504,0,448,450)
calibration info updated.
```

Export done.   Database: localhost, mts400_results   MIB520: COM4@57600

Fig. 10: Recordings of the sensor values without node ID 5299


MoteView 2.0
File Settings Tools Units Help
LIVE ☑
Data Command Charts Health Histogram Scatterplot Topology
Show sensor and node data in a spreadsheet format
5299
5315
GW
5304
lightx : 89.93Lux
lightx : 478.17Lux
5312
lightx : 411.93Lux
5318
lightx : 478.17Lux
5325
lightx
High
Low
Sensor Network Topology
Crossbow Technology, Inc. Copyright 2006
start MoteView 2.0 MoteView_UserManu... new images 2:35 PM

Id	Name
00	Gateway
5299	Node 5299
5304	Node 5304
5312	Node 5312
5315	Node 5315
5318	Node 5318
5325	Node 5325

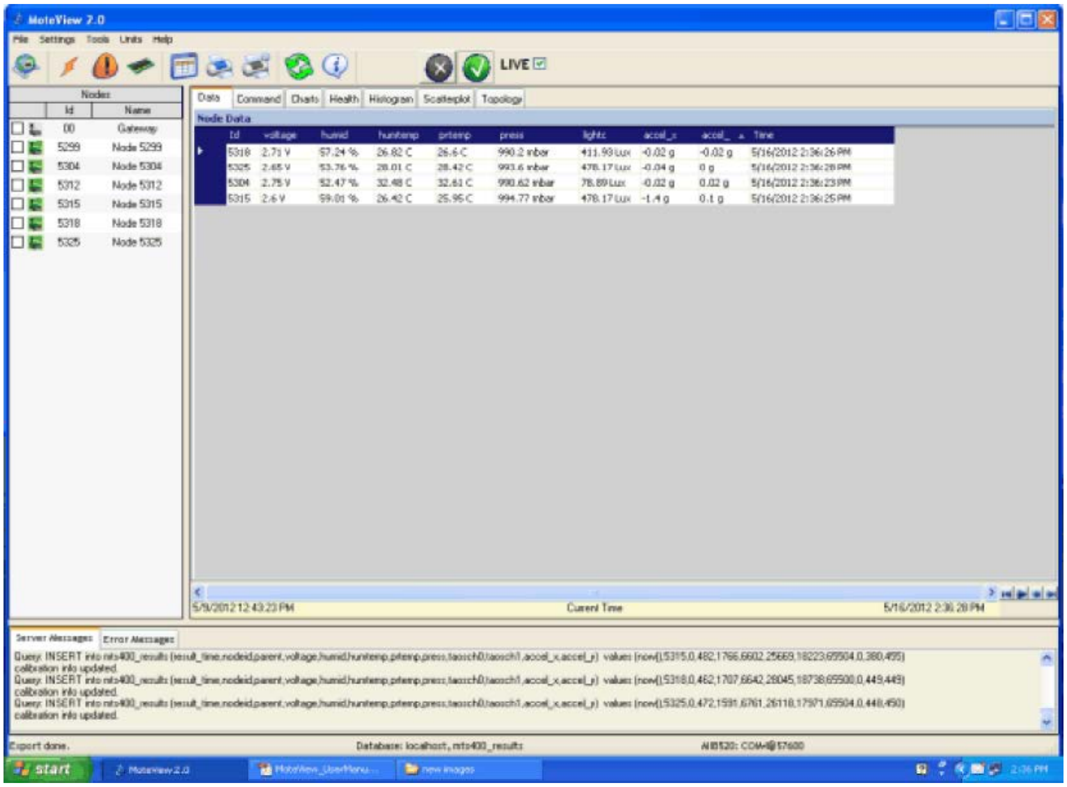
5/9/2012 12:43:23 PM Current Time 5/16/2012 2:35:28 PM

Server Messages | Error Messages

```
Query: INSERT into mts400_results (result_time,nodeid,parent,voltage,humid,humtemp,prtemp,press,taosch0,taosch1,accel_x,accel_y)  values (now(),5315,0,482,1810,6558,29595,18234,65504,0,381,455)
calibration info updated.
Query: INSERT into mts400_results (result_time,nodeid,parent,voltage,humid,humtemp,prtemp,press,taosch0,taosch1,accel_x,accel_y)  values (now(),5318,0,461,1741,6618,27999,18745,65500,0,449,449)
calibration info updated.
Query: INSERT into mts400_results (result_time,nodeid,parent,voltage,humid,humtemp,prtemp,press,taosch0,taosch1,accel_x,accel_y)  values (now(),5325,0,472,1626,6744,26088,17976,65504,0,448,450)
calibration info updated.
```

Export done.   Database: localhost, mts400_results   MIB520: COM4@57600

Fig. 11: Topology represents that node ID 5299, ID 5325 has removed (captured) from the network


MoteView 2.0
File Settings Tools Units Help
LIVE ☑
Data Command Charts Health Histogram Scatterplot Topology
Node Data
Current Time
Server Messages Error Messages
start MoteView 2.0 MoteView_UserManu... new images 2:36 PM

Id	Name
00	Gateway
5299	Node 5299
5304	Node 5304
5312	Node 5312
5315	Node 5315
5318	Node 5318
5325	Node 5325

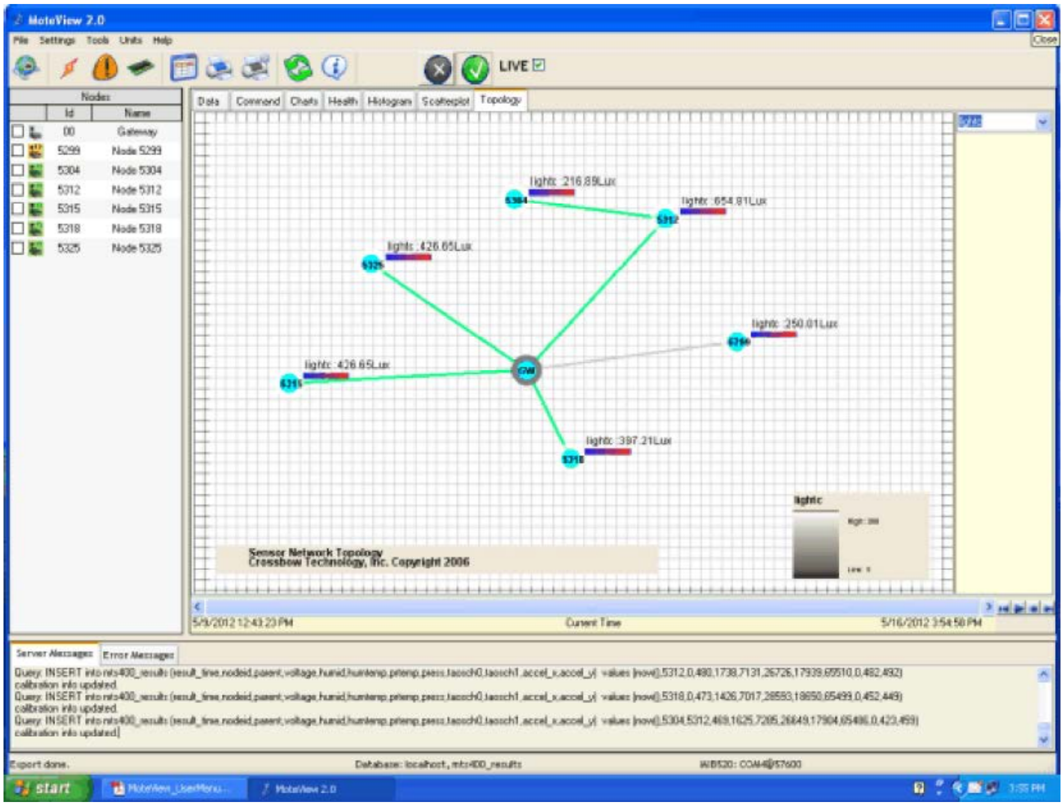
Id	voltage	humid	humtemp	prtemp	press	lightc	accel_x	accel_	Time
5318	2.71 V	57.24 %	26.82 C	26.6 C	990.2 mbar	411.93 Lux	-0.02 g	-0.02 g	5/16/2012 2:36:26 PM
5325	2.65 V	53.76 %	28.01 C	28.42 C	993.6 mbar	478.17 Lux	-0.04 g	0 g	5/16/2012 2:36:28 PM
5304	2.75 V	52.47 %	32.48 C	32.61 C	990.62 mbar	78.89 Lux	-0.02 g	0.02 g	5/16/2012 2:36:23 PM
5315	2.6 V	59.01 %	26.42 C	25.95 C	994.77 mbar	478.17 Lux	-1.4 g	0.1 g	5/16/2012 2:36:25 PM

5/9/2012 12:43:23 PM Current Time 5/16/2012 2:36:28 PM

```
Query: INSERT into mts400_results (result_time,nodeid,parent,voltage,humid,humtemp,prtemp,press,taosch0,taosch1,accel_x,accel_y) values (now(),5315,0,482,1766,6602,25669,18223,65504,0,380,495)
calibration info updated.
Query: INSERT into mts400_results (result_time,nodeid,parent,voltage,humid,humtemp,prtemp,press,taosch0,taosch1,accel_x,accel_y) values (now(),5318,0,462,1707,6642,28045,18738,65500,0,449,449)
calibration info updated.
Query: INSERT into mts400_results (result_time,nodeid,parent,voltage,humid,humtemp,prtemp,press,taosch0,taosch1,accel_x,accel_y) values (now(),5325,0,472,1591,6761,26118,17971,65504,0,448,450)
calibration info updated.
```

Export done. Database: localhost, mts400_results MIB520: COM4@57600

Fig. 12: Recordings of the sensor values without nodes ID 5299 and ID 5325


MoteView 2.0
File Settings Tools Units Help
Close
LIVE ☑
Data Command Charts Health Histogram Scatterplot Topology
lightc :216.89Lux
5304
lightc :654.81Lux
5312
lightc :426.65Lux
5325
lightc :250.01Lux
5299
lightc :426.65Lux
5315
GW
lightc :397.21Lux
5318
lightc
High
Low
Sensor Network Topology
Crossbow Technology, Inc. Copyright 2006
Current Time
Server Messages Error Messages
start MoteView_UserManu... MoteView 2.0 3:55 PM

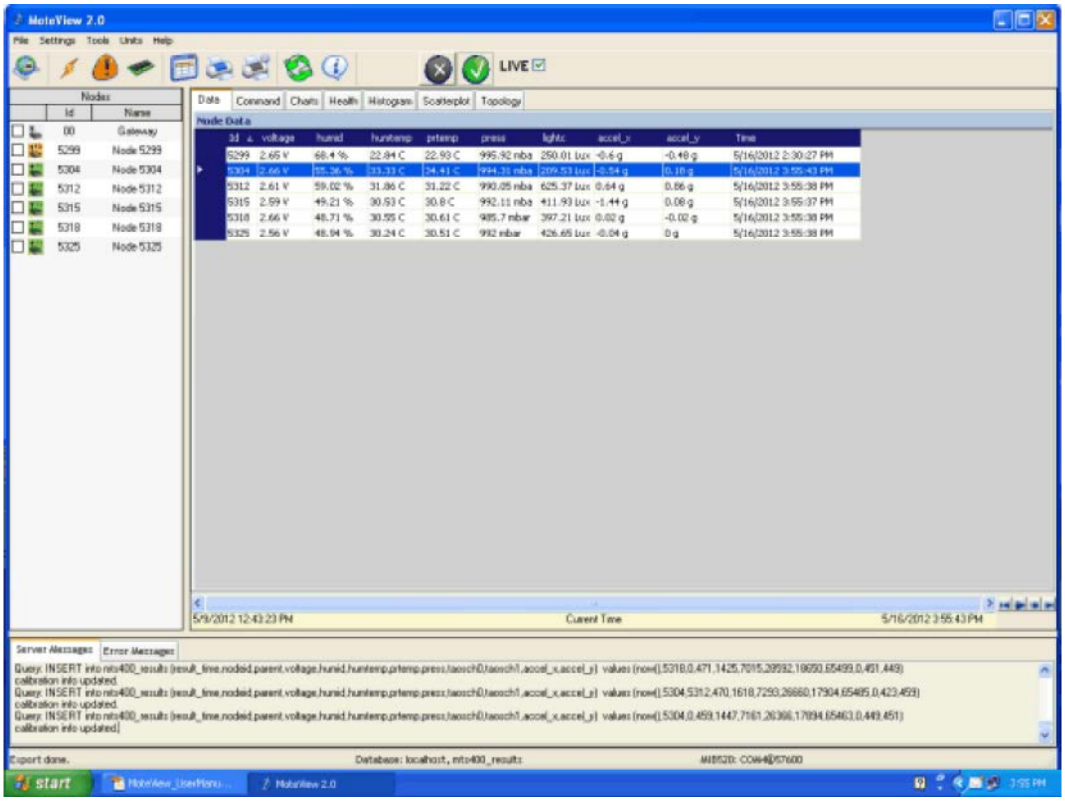
Id	Name
00	Gateway
5299	Node 5299
5304	Node 5304
5312	Node 5312
5315	Node 5315
5318	Node 5318
5325	Node 5325

5/9/2012 12:43:23 PM Current Time 5/16/2012 3:54:58 PM

```
Query: INSERT into mts400_results (result_time,nodeid,parent,voltage,humid,humtemp,prtemp,press,taosch0,taosch1,accel_x,accel_y) values (now(),5312,0,480,1738,7131,26726,17939,65510,0,482,492)
calibration info updated.
Query: INSERT into mts400_results (result_time,nodeid,parent,voltage,humid,humtemp,prtemp,press,taosch0,taosch1,accel_x,accel_y) values (now(),5318,0,473,1426,7017,28593,18650,65499,0,452,449)
calibration info updated.
Query: INSERT into mts400_results (result_time,nodeid,parent,voltage,humid,humtemp,prtemp,press,taosch0,taosch1,accel_x,accel_y) values (now(),5304,5312,469,1625,7205,26849,17904,65486,0,423,459)
calibration info updated.
```

Export done. Database: localhost, mts400_results MIB520: COM4@57600

Fig. 13: Topology shows that multi hop communication


MoteView 2.0
File Settings Tools Units Help
LIVE ☑
Data Command Charts Health Histogram Scatterplot Topology
start MoteView_UserManu... MoteView 2.0 3:55 PM

	Id	Name
□	00	Gateway
□	5299	Node 5299
□	5304	Node 5304
□	5312	Node 5312
□	5315	Node 5315
□	5318	Node 5318
□	5325	Node 5325

Node Data

Id	voltage	humid	humtemp	prtemp	press	lightc	accel_x	accel_y	Time
5299	2.65 V	68.4 %	22.84 C	22.93 C	995.92 mba	250.01 lux	-0.6 g	-0.48 g	5/16/2012 2:30:27 PM
5304	2.66 V	55.36 %	33.33 C	34.41 C	994.31 mba	209.53 lux	-0.54 g	0.18 g	5/16/2012 3:55:43 PM
5312	2.61 V	59.02 %	31.86 C	31.22 C	990.05 mba	625.37 lux	0.64 g	0.86 g	5/16/2012 3:55:38 PM
5315	2.59 V	49.21 %	30.53 C	30.8 C	992.11 mba	411.93 lux	-1.44 g	0.08 g	5/16/2012 3:55:37 PM
5318	2.66 V	48.71 %	30.55 C	30.61 C	985.7 mbar	397.21 lux	0.02 g	-0.02 g	5/16/2012 3:55:38 PM
5325	2.56 V	48.94 %	30.24 C	30.51 C	992 mbar	426.65 lux	-0.04 g	0 g	5/16/2012 3:55:38 PM

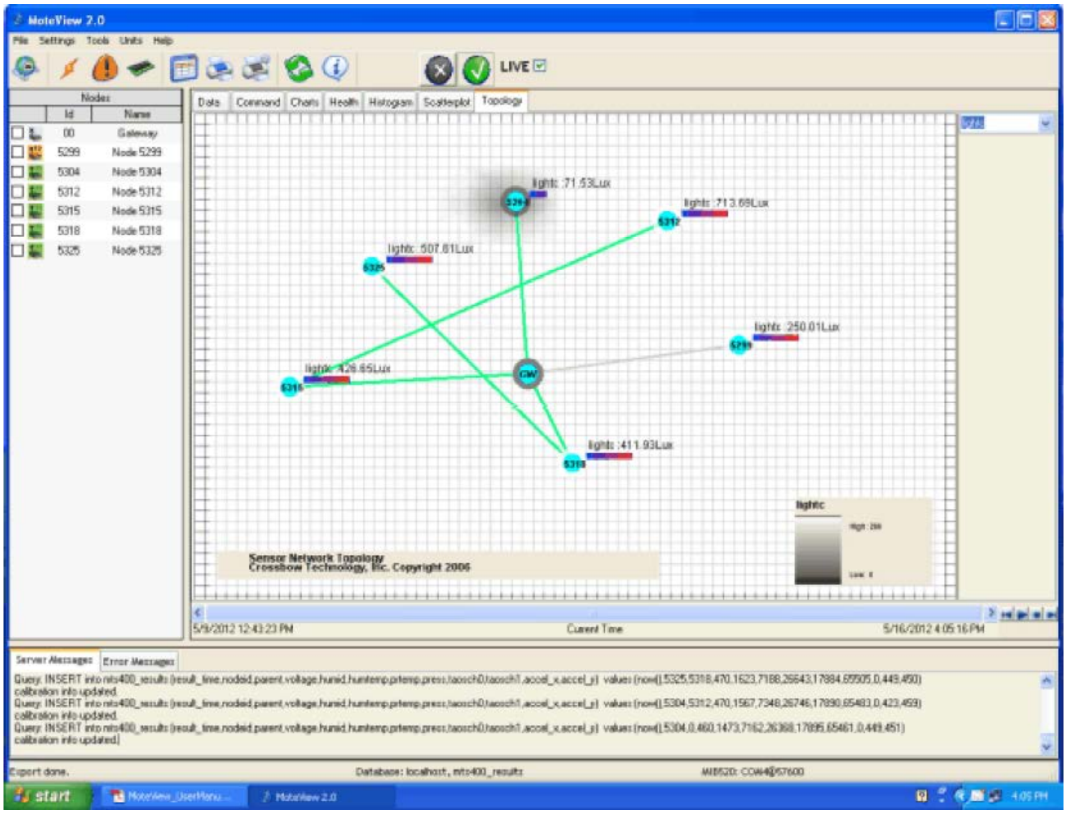
5/9/2012 12:43:23 PM Current Time 5/16/2012 3:55:43 PM

Server Messages Error Messages

```
Query: INSERT into mts400_results (result_time,nodeid,parent,voltage,humid,humtemp,prtemp,press,taosch0,taosch1,accel_x,accel_y)  values (now(),5318,0,471,1425,7015,28592,18650,65499,0,451,449)
calibration info updated.
Query: INSERT into mts400_results (result_time,nodeid,parent,voltage,humid,humtemp,prtemp,press,taosch0,taosch1,accel_x,accel_y)  values (now(),5304,5312,470,1618,7293,26660,17904,65485,0,423,459)
calibration info updated.
Query: INSERT into mts400_results (result_time,nodeid,parent,voltage,humid,humtemp,prtemp,press,taosch0,taosch1,accel_x,accel_y)  values (now(),5304,0,459,1447,7161,26366,17894,65463,0,449,451)
calibration info updated.
```

Export done. Database: localhost, mts400_results MIB520: COM4@57600

Fig. 14: Recordings shows that replicated ID overlap the values in the trusted ID field



	Id	Name
□	00	Gateway
□	5299	Node 5299
□	5304	Node 5304
□	5312	Node 5312
□	5315	Node 5315
□	5318	Node 5318
□	5325	Node 5325

5/9/2012 12:43:23 PM Current Time 5/16/2012 4:05:16 PM

Server Messages Error Messages

```
Query: INSERT into mts400_results (result_time,nodeid,parent,voltage,humid,humtemp,prtemp,press,taosch0,taosch1,accel_x,accel_y)  values (now(),5325,5318,470,1623,7188,26643,17884,65505,0,449,450)
calibration info updated.
Query: INSERT into mts400_results (result_time,nodeid,parent,voltage,humid,humtemp,prtemp,press,taosch0,taosch1,accel_x,accel_y)  values (now(),5304,5312,470,1567,7348,26746,17890,65483,0,423,459)
calibration info updated.
Query: INSERT into mts400_results (result_time,nodeid,parent,voltage,humid,humtemp,prtemp,press,taosch0,taosch1,accel_x,accel_y)  values (now(),5304,0,460,1473,7162,26368,17895,65461,0,449,451)
calibration info updated.
```

Export done. Database: localhost, mts400_results MIB520: COM4@57600

Fig. 15: Topology shows that two nodes try to communicate with gateway by multi hop communication

Table 3: Notations Used

Notation	Significance
n	Number of nodes in the network
d	Average degree of each node
g	Number of witness node
p	Probability a neighbor will replicate location information
q	Probability the distance will match exactly between the coordinator and examine nodes in the network

Table 4: Summary of protocol cost

Protocols	Communication	Memory
Broadcast	O(n^2)	O(d)
Line-Selected Multicast	O(nvn)	O(vn)
Randomized Multicast	O(n^2)	O(vn)
Deterministic Multicast	O(gvn)	O(g)
Randomized Efficient Distributed Multicast	O(g.p.d.vn)	O(g.p.d)
Distance Vector Method	O(q.vn)	O(vn)

Table 5: Summary of protocol cost for n=6, d=2, g=1, p=1,q=1

Protocols	Communication	Memory
Broadcast	O(n^2) = 36	O(d) = 2
Line-Selected Multicast	O(nvn) = 14.69	O(vn) = 2.44
Randomized Multicast	O(n^2) = 36	O(vn) = 2.44
Deterministic Multicast	O(gvn) = 2.44	O(g) = 1
Randomized Efficient Distributed Multicast	O(g.p.d.vn) = 4.89	O(g.p.d) = 2
Distance Vector Method	O(q.vn) = 2.44	O(q) = 1

Similarly Fig. 12 shows the recordings of the sensor nodes which exclude the above said two captured nodes. Then the replicated nodes continuously record the values from the place of trusted IDs invariably with the time intervals.

Fig. 13 shows how the replicated node tries to enter into the network through the trusted node in a multi hop way and Fig. 14 shows the corresponding reading of the above said scenario.

Similarly Fig. 15 shows that a single node was captured from the network. It also shows that two nodes communicated to the coordinator through some other nodes available nearby in multi hop way.

Performance Analysis: Extensive simulations were carried out using distance vector based method for identifying node replication attacks. Table 3, shows the notations used in different existing methods to compute the communication and memory cost.

Table 4 illustrates memory and communication costs for each protocol. From the same, the analysis was made to show how distance vector based method is efficient than other existing protocols. There is significant change in communication cost and the overhead of memory remains the same with line-selected multicast and deterministic multicast methods. In distance vector method there is no need to consider the average degree of each node (neighbor nodes),'d'. Only the considered fact is 'q' which says the probability that the distance will match exactly between the coordinator and examine nodes in the network.

Fig. 16 shows the comparison of distance vector method with already existing methods mentioned in the related works. For the sample consider n=6, d=2, g=1, p=1 and maximum probability q=1. After substituting all the results listed in Table 5, it shows that deterministic multicast and distance vector methods are equally better than all the other methods in terms of communication and memory cost. But comparatively a distance vector method is better than the deterministic method because there is no need of witness node here. The absence of witness nodes will improve the speed and other computational processes of finding the replication nodes present in the network.

The following graphs Fig. 16 and Fig. 17 shows the Communication and Memory overheads for the simulated results. It shows both deterministic and distance vector methods yields the efficient result in terms of communication and memory and among them the distance vector method has less computational overhead due to the absence of witness node consideration.

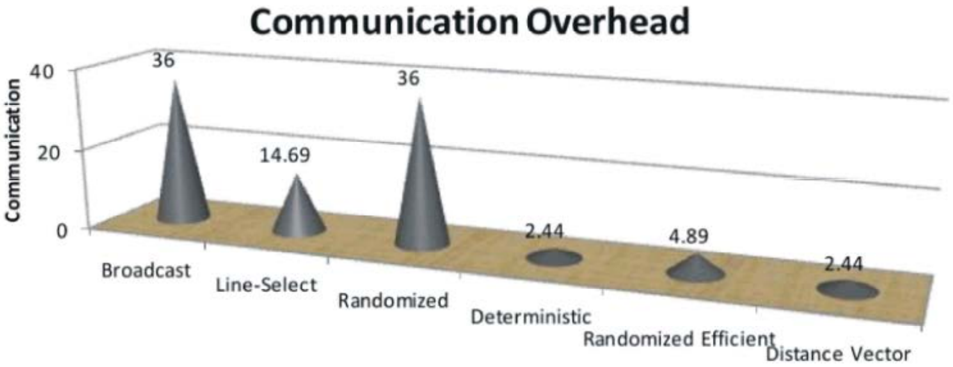


Fig. 16: Communication overhead for the values n=6, d=2, g=1, p=1, q=1

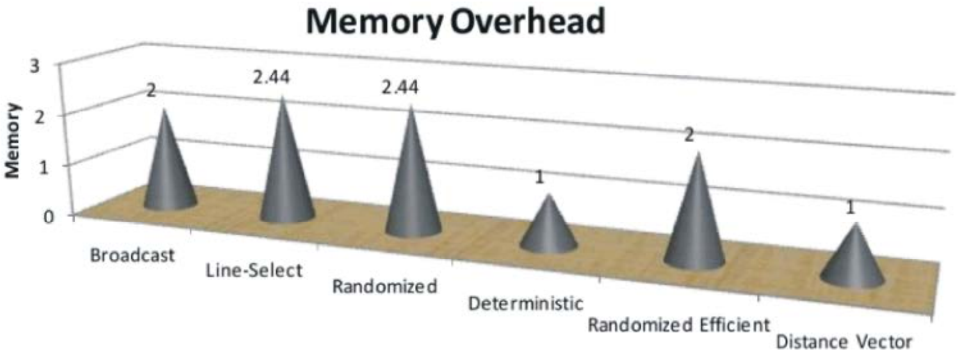


Fig. 17: Memory overhead for the values n=6, d=2, g=1, p=1, q=1

CONCLUSION

In this paper we propose a Distance Vector Based Detection (DVBD) of Node Replication Attacks in Wireless Sensor Networks using Haversine Method. The theoretical analysis and empirical results shows DVBD achieve higher probability of detection and lower communication costs than the previous schemes. Even if some nodes are compromised, DVBD still provides higher probability of detection. Extensive simulations confirm these results. Furthermore, DVBD achieves near real-time detection of node replication attacks. In the future, we would like to do more experiments, including time as a primary parameter to detect replicated nodes and comparisons with previous related works in terms of communication and memory cost.

REFERENCES

1. Mauro Conti, Luigi Vincenzo Mancini and Alessandro Mei, 2011. Distributed Detection of Clone Attacks in Wireless Sensor Networks, IEEE Transactions on Dependable and Secure Computing, 8(5).
2. Eschenauer L. and V.D. Gligor, 2002. A Key-Management Scheme for Distributed Sensor Networks, Proc. Conf. Computer and Comm. Security (CCS ’02), pp: 41-47.
3. Brooks R., P. Govindaraju, M. Pirretti, N. Vijaykrishnan and M.T. Kandemir, 2007. On the Detection of Clones in Sensor Networks Using Random Key Predistribution, IEEE Trans. Systems, Man and Cybernetics, Part C: Applications and Rev., 37(6): 1246-1258.
4. Bekara, C. and M. Laurent-Maknavicius, 2007. A new protocol for securing wireless sensor networks against nodes replication attacks, In Proceedings of the 3rd IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob).
5. Parno, B., A. Perrig and V.D. Gligor, 2005. Distributed Detection of Node Replication Attacks in Sensor Networks, Proceedings IEEE Symposium, Security and Privacy, pp: 49-63.
6. Menezes, A.J., S.A. Vanstone and P.C.V. Orschot, 1996. Handbook of applied cryptography, CRC Press, Inc.

7. Zhu, B., V.G.K. Addada, S. Setia, S. Jajodia and S. Roy, 2007. Efficient distributed detection of node replication attacks in sensor networks, In Proceedings of the 23rd Annual Computer Security Applications Conference (ACSAC).
8. Ratnasamy, S., B. Karp, L. Yin, F. Yu, D. Estrin, R. Govindan and S. Shenker, 2002. GHT: A geographic hash table for data-centric storage, In Proceedings of the 1st ACM International Conference on Wireless Sensor Networks and Applications (WSNA).
9. Heesook Choi, Sencun Zhu and T.F. La Porta, 2007. SET: Detecting node clones in Sensor Networks, In Proceedings of the 3rd International Conference on Security and Privacy in Communication Networks (SecureComm).
10. Bekara, C. and M. Laurent-Maknavicius, 2007. A new protocol for securing wireless sensor networks against nodes replication attacks, In Proceedings of the 3rd IEEE International Conference on Wireless and Mobile Computing,Networking and Communications (WiMob).
11. Xing, K., F. Liu, X. Cheng and D.H.C. Du, 2008. Real-time detection of clone attacks in wireless sensor networks, In Proceedings of the 28th International Conference on Distributed Computing Systems (ICDCS).
12. Xing, K., X. Cheng, L. Ma and Q. Liang, 2007. Superimposed code based channel assignment in multi-radio multi-channel wireless mesh networks, In Proceedings of the 13th Annual International Conference on Mobile Computing and Networking (MobiCom).
13. Conti Mauro, Luigi Vincenzo Mancini and Alessandro Mei, 2011. Distributed Detection of Clone Attacks in Wireless Sensor Networks, IEEE Transactions on Dependable and Secure Computing, 8(5).
14. Anandkumar, K.M. and C. Jayakumar, 2012. Prevention of Clone Attacks in Pervasive Health Care Environments, In transactions of European Journal of Scientific Research, ISSN 1450-216X, 72(3): 348-359.