Securing Wireless Sensor Networks from Node Clone Attack

Project report submitted in fulfillment of the requirement for the degree of Bachelor of Technology

In

Computer Science and Engineering

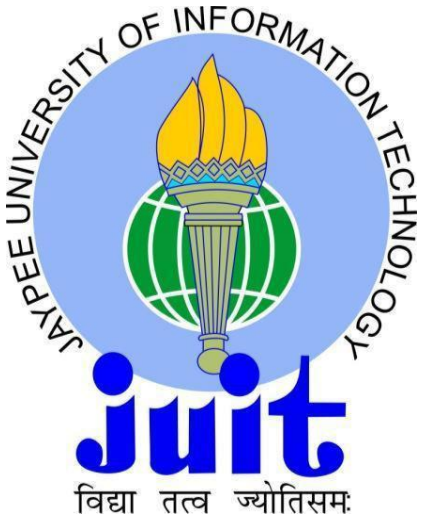
By

Adhishri Kothiyal (131290)

Under the supervision of

(Dr. Yashwant Singh)

To



Department of Computer Science & Engineering and Information Technology

Jaypee University of Information Technology Waknaghat, Solan-173234,

Himachal Pradesh

TABLE OF CONTENTS

CERTIFICATE	**II**
ACKNOWLEDGEMENT	**III**
ABSTRACT	**IV**
CHAPTER-1 INTRODUCTION	**01**
CHAPTER-2 LITERATURE SURVEY	**20**
CHAPTER-3 SYSTEM DESIGN	**30**
CHAPTER-4 PERFORMANCE ANALYSIS	**40**
CHAPTER-5 CONCLUSIONS	**50**
REFERENCES	**60**
APPENDICES	**70**

CERTIFICATE

Candidate's Declaration

This is to certify that the work which is being presented in the report entitled **"Securing from Node Clone attack in WSN"** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering/Information Technology** submitted in the department of Computer Science & Engineering and Information Technology**,** Jaypee University of Information Technology Waknaghat is an authentic record of our own work carried out over a period from August 2016 to May 2017 under the supervision of **Dr. Yashwant Singh** (Assistant Professor, Computer Science & Engineering Department).

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

Adhishri Kothiyal, 131290

This is to certify that the above statement made by the candidates is true to the best of my knowledge.

Dr. Yashwant Singh
Assistant Professor

Computer Science & Engineering Department
Dated:

ACKNOWLEDGEMENT

Known words become inadequate to express our gratitude for our mentor and guide Dr. Yashwant Singh, Assistant professor in Department of Computer Science and Information Technology Engineering, who initiated us into the realm of research and supervised us with finite patience and without whose invaluable suggestion and unstinted co-operation, the present desertion would not have been possible. Also, we would like to thank the officials of Jaypee University of Information Technology, Waknaghat for their help and cooperation.

Signature of the student ……..

Name of Student Adhishri Kothiyal

Date ……..

ABSTRACT

WSN are networks of small computing devices that can sense data and this data can be useful for some important job. WSN are used in many areas like battle surveillance, zebra monitoring system, sniper detection system and health monitoring systems. Since these networks susceptible to a variety of attacks, so it becomes necessary to secure these networks from such attacks. A central problem in Wireless Sensor Networks is that the nodes are susceptible to physical attacks. Once a sensor is compromised, attacker can easily launch a node clone attack by replicating the compromised nodes, distributing these in the network and then a variety of insider attacks can be launched. Previous works against the node clone attack suffer either from high communication cost or from poor detection accuracy. In this thesis, we are proposing a system that detects node clones assuming that WSN to be static and using a distributed scheme. The contribution of this paper is twofold. First, we studied the traditional methods for the node clone detection and discuss their drawbacks. Second, we implement an algorithm that will eliminate the drawbacks in the traditional schemes and prove its efficiency using simulations.

CHAPTER-1

1.1 INTRODUCTION

Wireless sensor networks are networks consisting of small low cost computing devices that can sense data from the environment, process the sensed data locally and then by using multi-hop relaying send the data to sink which is usually connected to other networks through a gateway. They comprise of cutting edge organize designs and in this manner are utilized as a part of a wide assortment of uses [1][2].

The WSN comprises of “nodes". Each “node” has four components which are as follows: a “radio transceiver” which is connected to an antenna, a “microcontroller”, which acts as an interface. A “battery source” and to fourth and main component is the “sensor node” which can be quiet small or big depending on the need. Extremely small "motes" are not developed till date.

WSN comprises of “nodes" which can vary from few to a large number. Every “node” is attached to different number sensors. Wireless sensor networks is defined as “ a collection of spatially distributed autonomous sensors to *monitor* physical or environmental conditions, such as temperature, sound, pressure, etc. and to cooperatively pass their data through the network to a main location.”

WSN is bi-directional. It checks the functioning of all the other sensors present in a network. Wireless sensor networks applications - to notice animals conduct in remote areas like in a jungle, to notice temperature in an area after a fire and then “temperature map” is created, in smart buildings, in intensive care in hospitals, etc.

It requires lesser cost, lesser manpower, lesser time, delivers data in real time and most importantly helps to develop a perfect water status graph . WSNs topology is one of the following -- star topology, wireless mesh topology or cross layered topology.

Attributes of WSN are:

It is resistant to failing of node; it is scalable; easy –to –use, has “cross layer” design, has “heterogeneous” nodes, is resistant to severe environment conditions and consumes less power comparatively.

Designing of WSN is cross layered now as traditional “layered approach” had many shortcomings that were as follows

1. It could not pass data among the various layers.

2. It did not assure network optimization
3. It could not modify in response to environment change.
4. It could not be used in wireless network.

Therefore nowadays cross layered approach is preferred for improving the transmission efficiency and the QoS.

Sensor nodes comprise a processing unit; radio transceivers and a power source like battery. Producing lower cost and smaller size sensor. has not been possible till now. Research is still going on this issue.

WSN performs communication with a "Local Area Network" or Wide Area Network" via gateway. The Gateway behaves like an interface for the Wireless Sensor Network and the network.

Wireless sensor network uses simpler OS as compared to normal OS used in our PC's. OS of WSN is similar to embedded systems. Wireless sensor networks are application specific. They do not have a general platform. Operating system like eCos and uC/OS are used for sensor networks.

WSN consists of following major components:-

Sensors

Each senor has covers some are area in WSN and senses the input data and then sends the sensed information to microcontroller for further processing. Input for a sensor can be temperature, light, heavy metals, gases, vapor, pressure, etc. Output is an analog signal which is further converted into digital signal using "ADC" convertor changed to human-readable form so that humans can read it.

There are many varieties of sensor each of which has a different cost and size. Wireless sensor nodes are small in size so they are supplied power which is lesser than 0.6-2 ampere-hour and 1.3-3.8 volts.

Sensors are of three types –"passive, omnidirectional sensors; passive, narrow-beam sensors; and active sensors".

- Passive sensors are self – powered.
- Active sensors are not self-powered.

Theoretical working on WSNs like in this project imagines using passive, omnidirectional sensors.

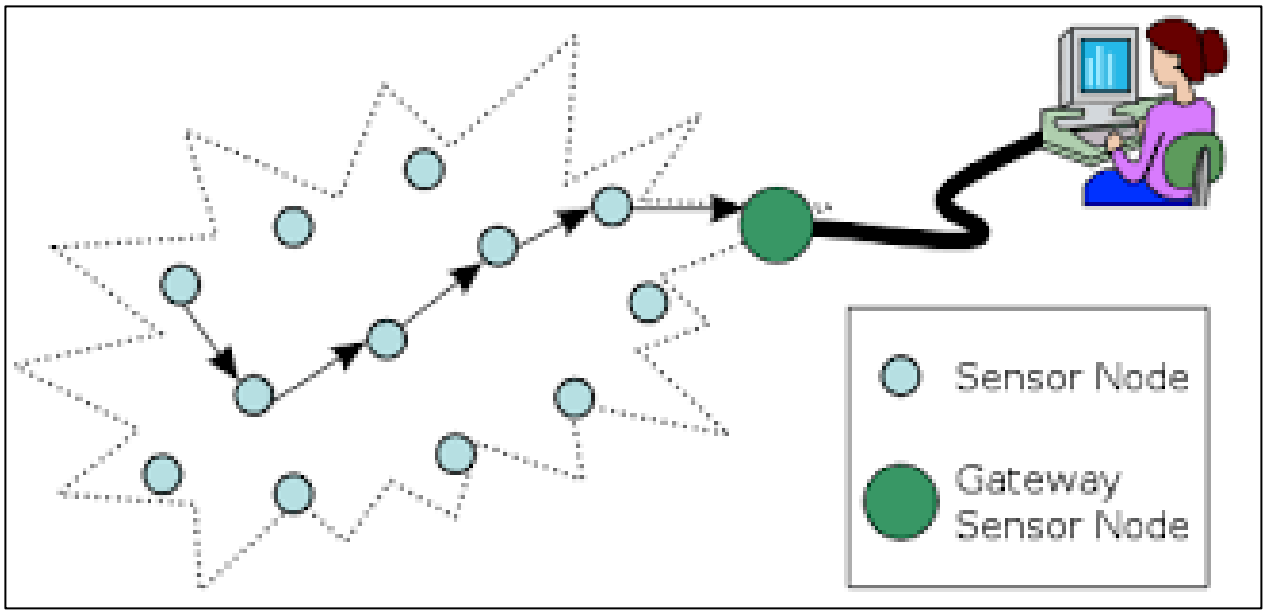


Figure1. Wireless Sensor Network [54]

Controller

Controller processes the input information and checks functioning of all the other parts in the node. Examples of controllers that can be used are "microcontroller", "digital signal processors", "Field Programming Gate Array", etc. A microcontroller is used due to its lesser cost. It is also very flexible in connecting to another device. It is easily programmable. It even consumes less power. Microprocessor uses high power in comparison to microcontroller, so a microcontroller is used most of the times.

Power Sources

Sensor node needs power supply for detecting data, communicating and then processing it. If it is not provided with required amount of power supply then it will stop working.

Mostly the wireless sensor node is present in a faraway location so using battery as a power source for it is not a good option as battery has to be changed on regular basis. Battery or capacitor is used as a power source for it .It could or could not require charging. Power in battery is saved in following ways – By switching off part of nodes that are presently not in use .This is called "Dynamic Power Management" or by changing amount of power supplied to nodes depending on situation This is called "Dynamic Voltage Scaling

External Memory

Sensors make use of flash memory or memory on chip because they are most energy efficient. Flash memory costs less and even has high storing capability. Memory required depends on the application being developed.

Sensor Nodes of Wireless Sensors

The WSN sensor node is the principle building piece of the created WSN framework model. Conduct of the WSN sensor node is decided by a microcontroller and a software program .It is fitted with sensor and microcontroller units. The information detected by the sensor goes through a molding circuit which conditions it so that it can be processed efficiently in the upcoming stage. At the next point of time after this the information will be given to the controller. The wireless sensor hub in this comprises of sensor unit and a microcontroller .WSN depends on adaptable, simple to-utilize hardware and software.

A sensor node called "mote" is a node that performs collecting, processing, and communication with all the nodes present in the network. A fact about a mote is that --- "A mote is a node but a node is not always a mote".

The sensor node consists of:

- A microcontroller,
- A transceiver,
- External memory,
- A power source,
- Sensors

Base Monitoring Station

The base station gets the information sent from the sensor nodes i.e. end gadgets and switches, remotely. Information that is got from the end gadget/device is sent to the PC and information got is shown utilizing the Graphical User Interface on the base observing/monitoring station. A Base station is-"a fixed point of communication for customer cellular phones on a carrier network". The base station connects with an antenna which does the task of receiving and transmitting the signals in concerned person. The base station receives the data sent from the sensor nodes i.e. end devices and routers, wirelessly. Data received from the end device nodes is sent to the computer and data received is displayed using the built GUI on the screen base monitoring station.

Transceiver:-

It does the work of receiver as well as of a receiver.

It is defined as -- "a device that can both transmit and receive communications, in particular a combined radio transmitter and receiver."

Examples of transceivers that can be used for water quality monitoring system using WSN are "radio frequency" transceivers, "optical communication" transceivers and "infrared" transceivers. Infrared, transceiver does not require any antenna. It can transmit to a lesser distance. "Radio frequency" transceiver is the most commonly used transceiver in WSN. WSNs work on following frequencies: 174, 434, 867 and 916 MHz and 2.5 GHz. Transceivers do not have different identifiers that are one issue with it. It has the following states -- transmit, receive, idle, and sleep.

Transceiver should be switched off when not in use as it consumes a lot of power even when it is not working. When a packet is to be transmitted by the transceiver a large amount of power is used.

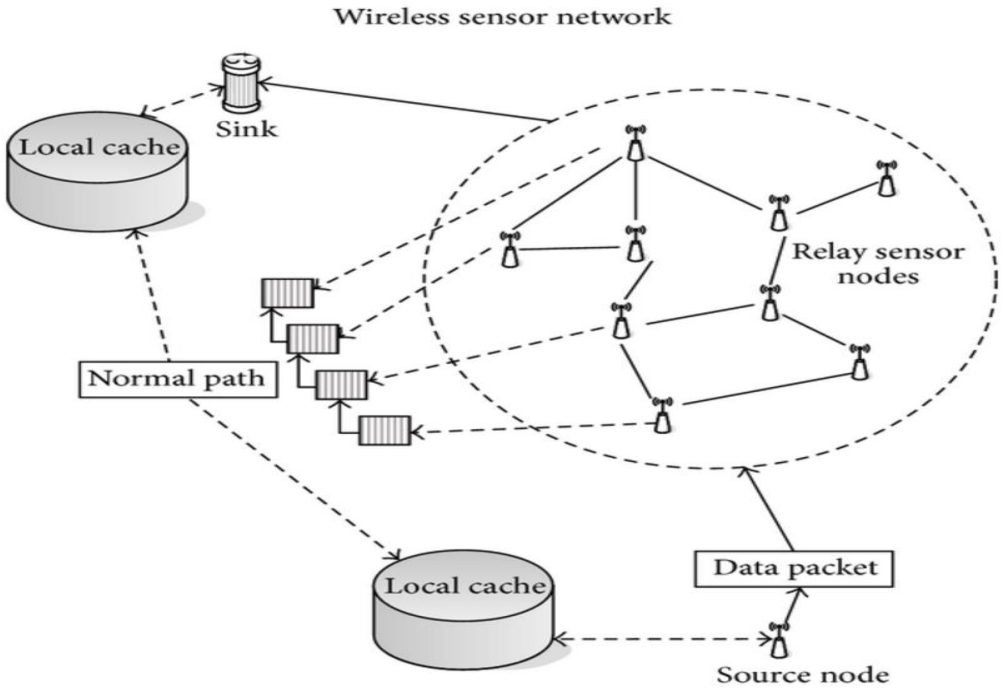


Figure2. Sensor Network [54]

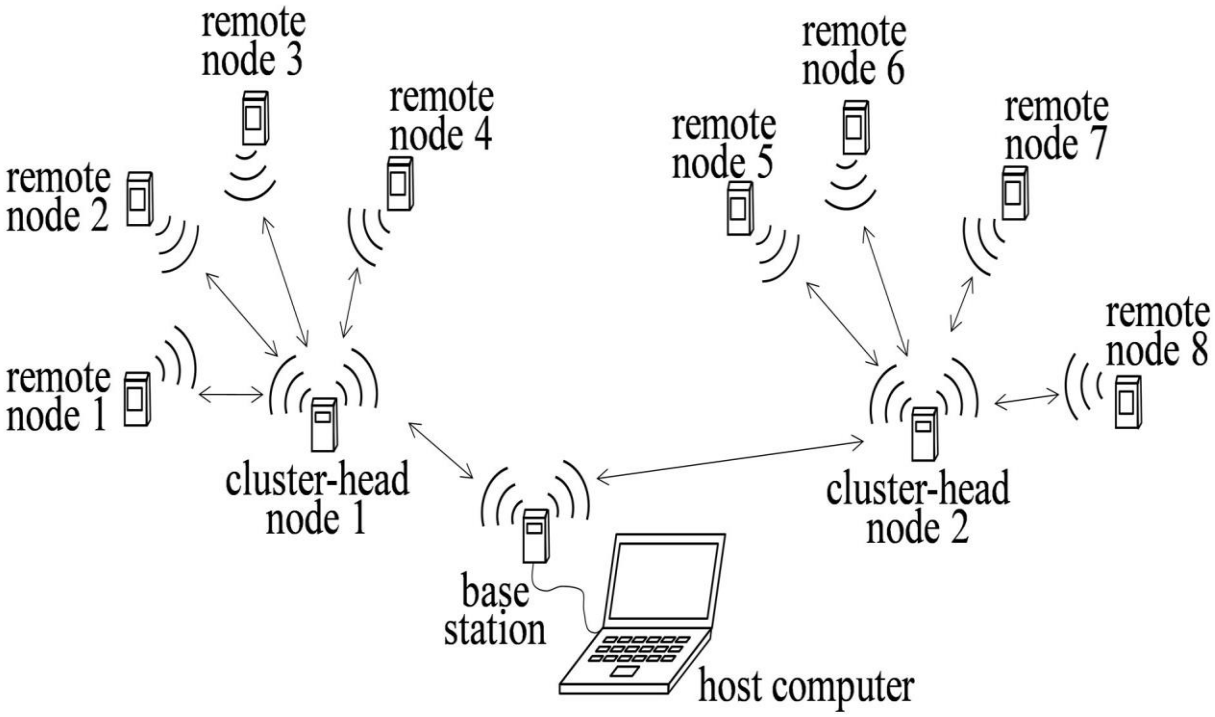


Figure3. Sensor Network Architecture [54]

1.2 APPLICATIONS OF WSN:

1. Sniper Detection system in which microphones is embedded on the shoulder of a soldier and then these sensors uses acoustic processing to calculate the trajectory of the incoming bullet and hence the position of sniper can be calculated.
2. Gas sensors networks are used to detect pollution levels in cities and a warning is triggered when the pollution levels become greater than the permissible limit.
3. Under-water sensors can detect and warn any disease beforehand and these are also used to detect under-water earthquakes.

Wireless Sensor Networks are prone to much type of attacks and these are:

Black-Hole attack: In this attack an attacker physically captures nodes and then reprograms them so that they do not forward any packet that comes from another node and thus whole network gets down.

Worm-hole attack: Wormhole nodes find a route that is shorter than the original node's route in the network; this confuses routing mechanisms which rely on the knowledge about distance between nodes. The attacking node captures the packets from one location and transmits them to other distant located node which distributes them locally. A wormhole attack can easily be launched by the attacker without having knowledge of the network or compromising any true nodes or cryptographic techniques.

Sinkhole attack: In a sinkhole attack, the opponent's aim is to attract nearly all the traffic from a particular area through a compromised node, creating a sinkhole with the adversary at the center. Sinkhole attacks work by making a compromised node look especially attractive to surrounding nodes with respect to the routing protocol. Sinkhole attacks are difficult to counter because routing information supplied by the node is difficult to verify. As an example, a laptop-class adversary has a strong power radio transmitter that allows it to provide high-quality route by transmitting with enough power to reach a wide area of the network.

Sybil attack: This attack is particularly confusing to geographic routing protocols as the antagonist appears to be in multiple locations at once. A malevolent node present various identities to the network is called Sybil attack.

Node Clone Attack: In this attack the attacker physically captures a node and then makes clones of the node using the node id and distributes these inside the network and therefore other attacks like DOS and black hole attack can be launched.

The organization of report is:

In First Section we will discuss some recent existing routing protocols and our problem statements and objectives.

Section II presents our methodology and literature survey which describes the network and adversary models used. This proposed method along with system design.

In Section III is the analysis part.

Section IV shows the results. At last in the end the report is concluded.

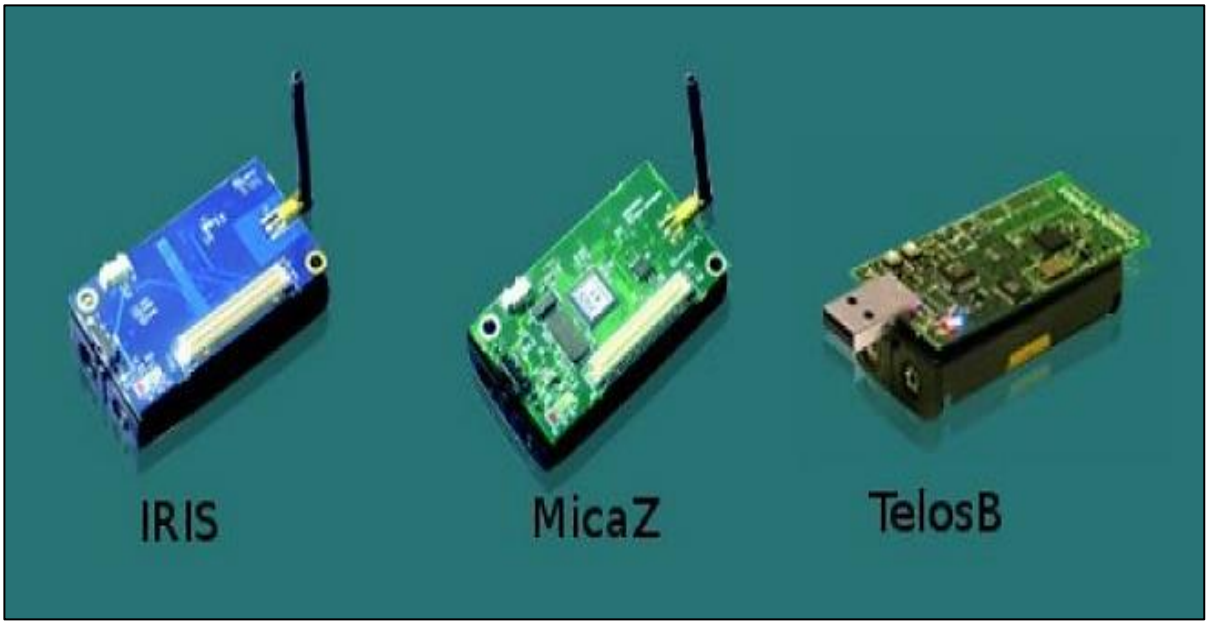


Figure4. Some practical nodes include Mica2, MicaZ, Telos, IRIS, etc. [54]

1.3 PROBLEM STATEMENT

The problem this paper aims to solve is the detection of node clones attack in WSN by studying various algorithms, protocols and discuss their drawbacks and finally, to simulate a system with an efficient algorithm.

1.4 OBJECTIVE

Short-term: The short-term objective of the project is to completely understand WSN, protocols used in them, different security issues and to analyze and compare different algorithms already given for detection of the node replication attack in WSN.

Long-term: Long-term objective of this project is to implement and simulate an efficient algorithm for the detection of clones.

1.5 METHODOLOGY

1. *Problem Statement*: First we studied about WSN networks and different types of attacks and then we studied Node clone attack in WSN.
2. Requirement Model: In this part, we study the requirements for an efficient distributed witness based protocols.

3. *Network and Adversary Model*: This section describes the assumed network and adversary models. The notations and symbols that we will be using in this report are also introduced.
4. *Data Collection*: After successfully studying about WSN and other attacks we analyzed different algorithm and compared their strengths and drawbacks.
5. *Protocol*: In this we describe the working of our protocol and analyzed various parameters' efficiency.
6. *Simulation*: Finally, we will simulate our threat model and then show how our proposed algorithm is successful.

CHAPTER-2

2.1 LITERATURE SURVEY

The first attempt for the detection of clones was a centralized one proposed in [39] which was a naïve solution relying on an assisted central authority or "base station assisted central authority".

The very first distributed solution (naïve) for node clone detection is "Node-to-Network Broadcasting" (N2NB) [24]. In N2NB a message is flooded by all the nodes containing the information of location in the network and after that the received information (of location) is compared with neighborhood nodes. A replica is identified on accepting a conflicting case. It is separated from the network after abortion. Some distributed approaches proposed to distinguish clone assaults by utilizing "claimer-reporter-witness framework" likewise called "witness node based technique" [24–32]. These are the most promising procedures up until this point. However there are still a few restrictions.

B.Parno ET. Al. [24] was the first to propose two probabilistic algorithms Line-Selected Multicast (LSM) and Randomized Multicast (RM) for the full fledge detection of clones/replicas in WSN which follow the claimer reporter witness method. In Randomized Multicast, when a claimer node announces its location by locally broadcasting the signed location claim to its neighbors, each of its neighbor nodes become a reporter with probability p after verifying the credibility of the location. Each reporter then selects random destinations in the network and forwards the genuine location claim to the node close to those haphazard locations that are called witness nodes. If there is a clone in the network and the reporters of that cloned node also select O random end points then by exploiting the birthday paradox at least one common witness will receive two conflicting location claims with high probability. This witness node can revoke the clone/replica node and immediately publicize the network with the evidence of incoherent location claims to discredit. Randomized Multicast means high communication cost as each neighbor has to send messages to accomplish common witnesses. In Line-Selected Multicast, when a claimer node declares its location, each neighbor becomes a reporter and has a probability p after locally checking the signature of the claim and then forwards it to randomly selected end points. The location claim must pass through several intermediate nodes during its propagation from a reporter node to a witness node, on the forwarding route which also store the location claim randomly drawing a line across the network and thus serve as additional witnesses. In case of a replicated node, when a clashing location claim by a clone node crosses the forwarded path for the legitimate node, the B.Parno ET. Al. [24] was

the first to propose two probabilistic calculations Line-Selected Multicast (LSM) and Randomized Multicast (RM) for the full fledge recognition of clones/copies in WSN which take after the claimer correspondent witness strategy. In Randomized Multicast, when a claimer hub declares its area by locally communicating the marked area claim to its neighbors, each of its neighbor hubs turn into a columnist with likelihood p subsequent to confirming the validity of the area. Every journalist then chooses arbitrary goals in the system and advances the veritable area claim to the hub near those indiscriminate areas that are called witness hubs. In the event that there is a clone in the system and the columnists of that cloned hub likewise select O arbitrary end focuses then by misusing the birthday Catch 22 no less than one basic witness will get two clashing area claims with high likelihood. This witness hub can repudiate the clone/reproduction hub and instantly expose the system with the proof of mixed up area cases to ruin. Randomized Multicast implies high correspondence taken a toll as each neighbor needs to send messages to fulfill regular witnesses. In Line-Selected Multicast, when a claimer hub proclaims its area, each neighbor turns into a columnist and has a likelihood p after locally checking the mark of the claim and afterward advances it to haphazardly choose end focuses. The area guarantee must go through a few moderate hubs amid its spread from a columnist hub to a witness hub, on the sending course which likewise store the area assert arbitrarily drawing a line over the system and in this manner fill in as extra witnesses.

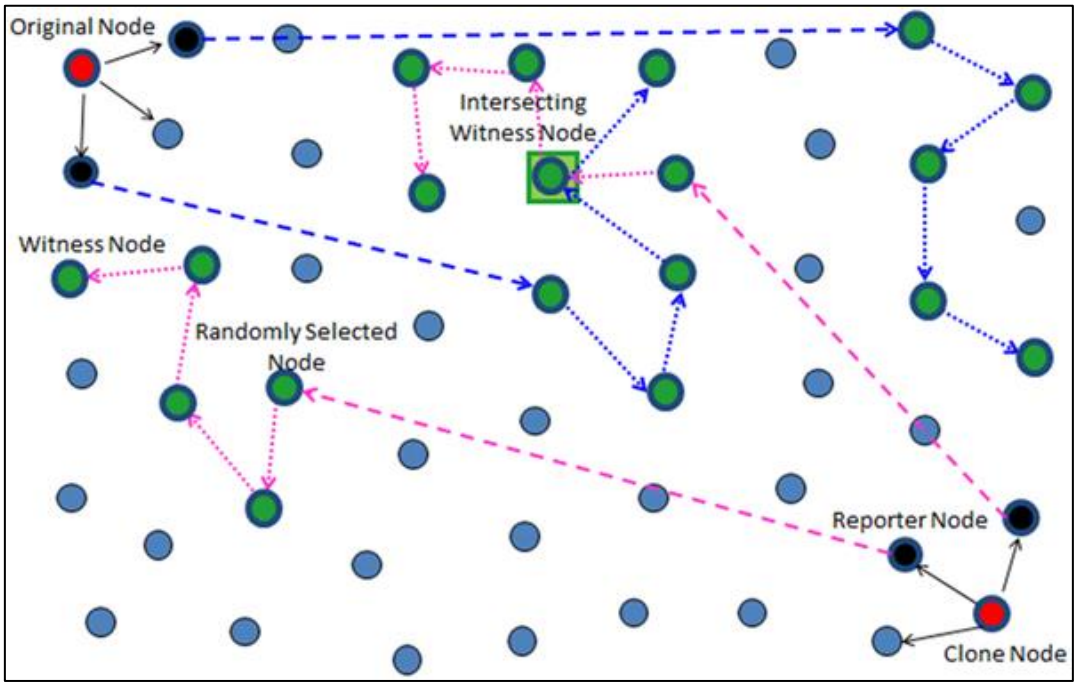


Figure5. Witness Node Selection

Steps of RAWL:-

1. Each node broadcasts a signed location claim.
2. Each of the node's neighbors forwards the claim probabilistically to some randomly selected nodes.
3. Each randomly selected node, containing the claim to start a random walk in the network, sends a message, and the witness nodes are selected by passed nodes and will store the claim.
4. If a different location claim is witnessed for a same node ID, it can use these claims to abort the cloned node.

Their 2nd protocol, TRAWL adds a trace table at every node to decrease memory cost and is based on RAWL. The RAWL needs more random walks steps and random walk in order to achieve a high detection probability and this leads to higher memory cost which is more than twice the communication overhead of LSM and higher communication cost.

Figure 5 shows how RAWL protocol works .The memory cost is reduced by proposing TRAWL but the communication cost still exist.

Both "RAWL and TRAWL" utilize comparative methodology for choosing "witness nodes", varying in that "TRAWL" lessening the costs of memory. This is done by employing trace table. This report, "RAWL" is chosen to compare with RWND, in light of the fact that first it is the most promising solution up until now, and secondly witness nodes are selected in RAWL by using random walks. In RWND we additionally utilize straightforward random walk however in a totally extraordinary way to select witnesses i.e. by merging system division with a novel witness choice technique.

In this report, a" review of commitment in [29] and after further exploring the "RWND" convention by hypothetically analyzing selection methods and the area division area, another mechanism for selection of witness node is presented. The examination and further simulations demonstrate that RWND outflanks the beforehand proposed conventions regarding high clone location probability and more grounded security of witness nodes along with direct overheads".

Alternate procedures for the discovery of node clone/replication assault in mobile and static sensor system can be found in more elaborated manner in [41–43].

CHAPTER-3

3.1 SYSTEM_DESIGN

3.1.1 Distributed witness node based method and its requirements

In witness node based techniques, basic witness (converging witness) nodes are critical element because these witnesses in the end distinguish and disavow the clones in the system. The security of these witness nodes is imperative as in deterministic conventions. It is comparatively simple for an attacker to catch and clone and after that compromise the witness nodes due to a small number of neighbors. An attacker can prevail by continuing the compromise of these witness nodes amid the lifetime of the network. To guarantee the security of witness node, the selection of these witnesses ought to be non-deterministic and every one of the node in the system ought to have an equivalent probability of being witnesses. Subsequently, it will be more troublesome for an enemy to effectively launch clone attack in non-deterministic protocol in light of the fact that the neighbors of a node are not known and they vary in every execution of the protocol. Additionally, the witness nodes ought to be consistently dispersed in the whole network. Also it must not be chosen more than once from a specific location of the network. These necessities ought to be satisfied to ensure the security of critical witness nodes which thus increases the detection likelihood of clones.

Since wireless sensor networks are resource constrained networks, both as far as energy of nodes and memory of node is concerned, it is in a way exceptionally difficult to design protocols with minimum overhead. In the event that the nodes begin depleting their batteries the entire system usefulness is disturbed. Additionally, if just few nodes are thought for high memory or capacity purposes then these nodes can flood which result about packet dropping or packet loss. This generously influences the discovery abilities of the protocol. Henceforth it is vital to develop such protocols which bring about normal or direct memory and overheads of communication while using the assets shrewdly and productively.

3.1.2 “Network and Adversary Models”

This section explains the assumed “network and adversary models” in detail. The notations and symbols used in the report are also introduced in this section.

3.1.2.1 Network Model

Consider a sensor network in which an extensive no. of cheap sensor hubs are consistently appropriated over a wide sending region. Every node is considered to know their own geographic areas by using some effectively exhibit limitation calculations. During the execution of clone detection protocol nodes are assumed to be stationary. Furthermore nodes are provided with a particularly specific personality with a couple of character based private and open keys. Same as past works [16–17] [24] [28] [29], we accept that foes can't make sensors with new personalities for clones as any two hubs are additionally thought to be ensured by combine shrewd keys. Keeping in mind the end goal to supplant the old sensor hubs new sensor hubs can be included into the system and like [25] [26] when another hub is included into the system, it needs to make an area guarantee. At that point this hub needs to communicate the claim to its neighbors.

3.1.2.2 Adversary Model

For a foe show, we accept an essential however viable foe that can first look and up some other time trade off the sensor hubs. By using cryptographic information acquired from those traded off hubs he makes reproductions/clones and later presents them in the system. We likewise accept the nearness of checking systems or robotized conventions like SWATT [44] which can draw human intervention and starts clearing the system to evacuate bargained hubs if an enemy tries to trade off limitless number of sensor hubs. Therefore, we accept that a foe may pick just a specific number of hubs to catch and trade off.

TABLE1. NOTATIONS USED

N_n	No. of sensor nodes in the network
N_a	No. of total areas in the network
N_{sa}	No. of selected areas by a reporter
N_c	No. of total combinations
I_a	No. of intersecting area(s)
S_a	Single selected area
d	Average degree of each node/ No. of neighbor of each node
P_{fd}	Probability of forwarding the location claim neighbor
P_d	Probability of detecting replica
r	No. of random walks for each node
||	Symbol for Concatenation
t	No.of walk steps by each random walk
loc_a	Location information of a node
ID_a	Node's unique Identity
K_a^{Pvt}	Private key of a node
K_a^{Pub}	Public key of a node
$Sig\{M\}_{Ka}{}^{pvt}$	Node a' signature on massage M
H(M)	Hash of massage M

3.1.2 "Random Walk with Network Division (RWND)"

This segment, a protocol called "Random Walk with Network Division (RWND)" for the discovery of replicas (*node clone attack*) is proposed. It is based on "claimer-reporter-witness framework".

3.1.2.1 Description

RWND comprises of random walk and the network division. RWND works in two phases. They are:-

I. *The phase of network configuration* and
II. *The phase of detection of clone*.

The phase of network configuration: - the whole network is divided into various levels. Afterwards these diverse levels define a specific area. Every node in the network falls in a particular area and level.

The phase of clone detection: - the replica is found by employing random walks within an area and following a "claimer-reporter-witness framework". Every node has to communicate a *signed location claim to* reporter (its neighbor hubs). Every reporter node likely forwards the claim to some randomly chosen hub from a blend of randomly selected areas. The neighbor(s) of a claimer node will choose a single node randomly in every territory. This node will further select additional *r* nodes randomly, that in the end will initiate the random walks. At each random walk step the passing nodes will become the witness nodes. Witness nodes in each area are randomly selected. They differ in every loop of this mechanism. The division of the system into direct measured territories grants a gigantic change over" RAWL" as far as memory cost and correspondence as the required number of arbitrary walk steps and irregular strolls are decreased. In addition to this, the "higher security of witness nodes is achieved by employing" and starting multiple parallel irregular strolls inside a blend of arbitrarily chose ranges. The formal depiction of each period of the method is explained in detail underneath.

I. The phase of network configuration

In this method, every hub in the whole network falls in a specific level w.r.t a specific sink (as there can be multiple sinks in any network, without any loss of generality, either one "sink or reference hub" is utilized) or any "reference hub". Here, the level speaks to the separation (jump forget about) to the doled sink and every zone includes an option number of levels, dependent upon the sink outline and arrangement. We expect that the level number in every region is static. It is arranged amid the period of system setup relying on the system estimate. "Partitioning the network into levels and areas is motivated by [45]".

The way toward tagging: - it incorporates division of network into different distinctive levels. At that point these levels are relegated to every one of the hubs. It is started dependably by the reference hub or sink. A message is sent by the sink hub to its one bounce neighbors and it contains zone/level and reference hub id/sink number. At the point when the message is gotten, each and every hub communicates a message to illuminate different hubs that it has a place with first level. The various hubs tuning in to this message and still not having this data increment the estimation of the gut level by one and afterward appoint themselves to this level and later check their region before communicating this new level. This procedure proceeds unless and until every one of the hubs have a place with a specific level and along these lines is doled out to a zone. When a hub has allotted itself to a specific level and range, it begins to disregard every single future communicate containing zone and level data. The above component of level and region task to hubs amid the period of system arrangement is appeared in Fig 5.

II. The phase of clone detection

By following the claimer-reporter-witness framework this phase works in 4 stages.

- Forwarding claim
- Selection of area,
- Selection of witness node
- Detection of clones and
- Revocation.

Forwarding Claim: This is the 1st stage. The process of detection begins. A" signed location claim" is forwarded by each node to its one hop node and becomes a "*Claimer Node"*. Syntax of the location claim is:" $\langle ID_a, loc_a, Sig\{H(ID_a||loc_a)\}_{K_a^{Pvt}}\rangle$"; here "||" depicts the method of concatenation. Also here "loc_a" is the info. of location of the node "*a"*.

Selection of Area: Subsequent to hearing the claim, each neighboring hub initially confirms the authenticity as well as location of the "claimer node" (example- separation amongst "claimer" and the one hop next node ought to be inside the range of transmission). The one hop next hub will turn into the "*Reporter Nodes"* hearing the claim, each neighboring center.

Two steps are performed reporter node(s).

In the initial which is called "*Area Selection*", reporter form a specific area forwards the location claim of a claimer node to some discretionarily picked areas using a mechanism of area selection. Now this part first portrays amount off zones that are picked by the feature writer node(s) from the total number of zones of the framework with the ultimate objective of sending the range. Any no. of zones has possible mixes which aren't ordered and do not have substitution. In wake of picking amount of reaches, the "reporter node" then subjectively chooses any 1 blend of districts for sending the territory ensure. "By taking after this framework, the reporter(s) from any scope of the framework can pick any mix of zones which realizes no short of what one intersection domain (typical zone)".

Selection of Witness Node: In "RAWL", the "witness center points" picked discretionarily the journalist node(s) which furthermore begin r discretionary walks around the whole framework took after by t unpredictable walk steps and a while later each passing center moreover transform into the witness centers. In our past work [29], whose working is in like manner showed up in Figure 4, the journalist node(s) discretionarily picks g (where g = r) geographic zones by using geographic guiding traditions (GPRS [46]) because of probability keeping the true objective to forward the claim to the g territories (according to [17], [28] picking a sporadic zone is vastly improved and more secure than picking Node ID) in each indiscriminately picked zone. In doing all things considered, disregarding the way that the required number of subjective walk steps was fundamentally diminished which in this manner diminishes the general correspondence cost to some degree when appeared differently in relation to RAWL due the division of the framework into

little areas yet the correspondence cost of editorialist to self-assertively picked center points had extended as the columnist needs to erratically pick r center points in each picked go for beginning the r unpredictable walks.

The 2nd step that every editorialist performs, portrays the "witness node selection" part that is overhauled in the report to fulfill more important security of "witness centers" and furthermore to decrease the correspondence expences.

This segment, reporter of a "*claimer center point*" will pick a single center self-assertively. Every picked domain each of which get the range attest and in the wake of affirming the stamp each of them will wind up being the witness center points in the wake of securing the territory ensure and will at long last start the r arbitrary strolls of t "walk steps", the hubs at every irregular "walk step" likewise turning into the neighborhubs. Figure demonstrates the enhanced witness hub determination instrument. This new witness center point decision framework restrains the cost of feature writer to self-assertively picked center in each range finally achieving reduced correspondence cost of our arrangement which is essential manner of thinking of this instrument. The reenactment happens affirm this method instrument of "witness center" decision has decreased the correspondence expensess for achieving greater acknowledgment liklihood when stood out from our past work and furthermore RWAL.

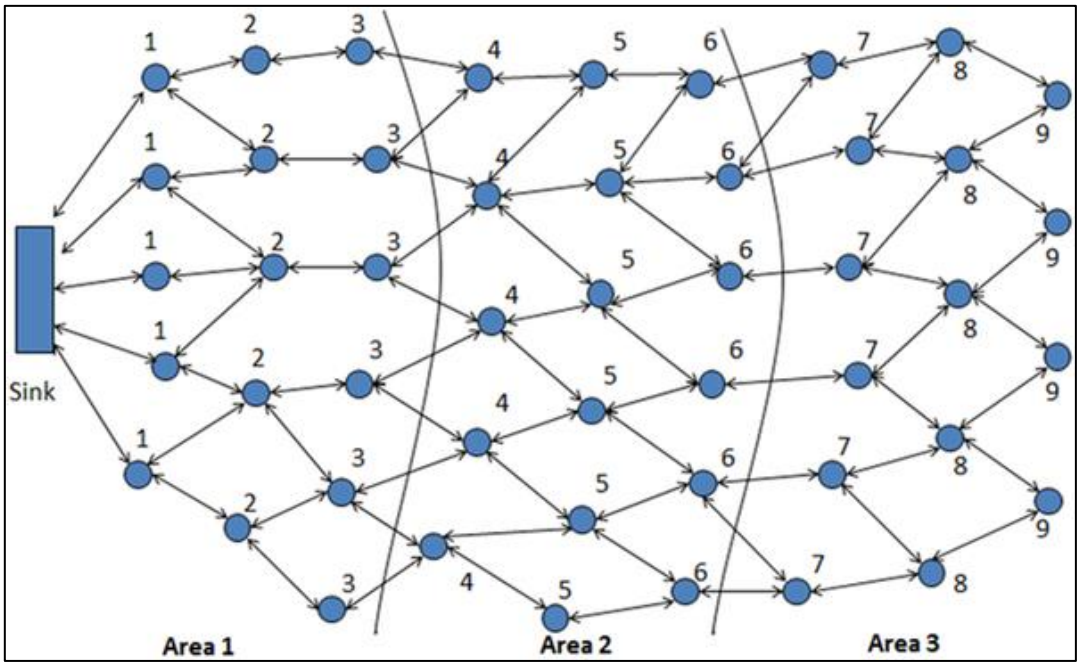


Figure6. Selection of Area []

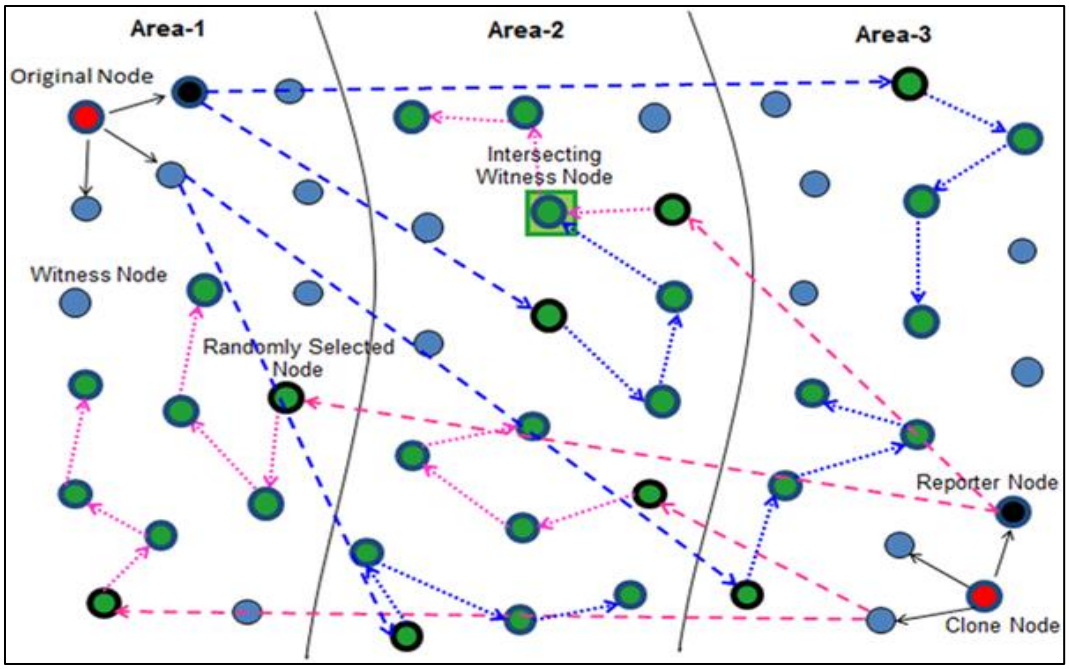


Figure7.Selection of witness node []

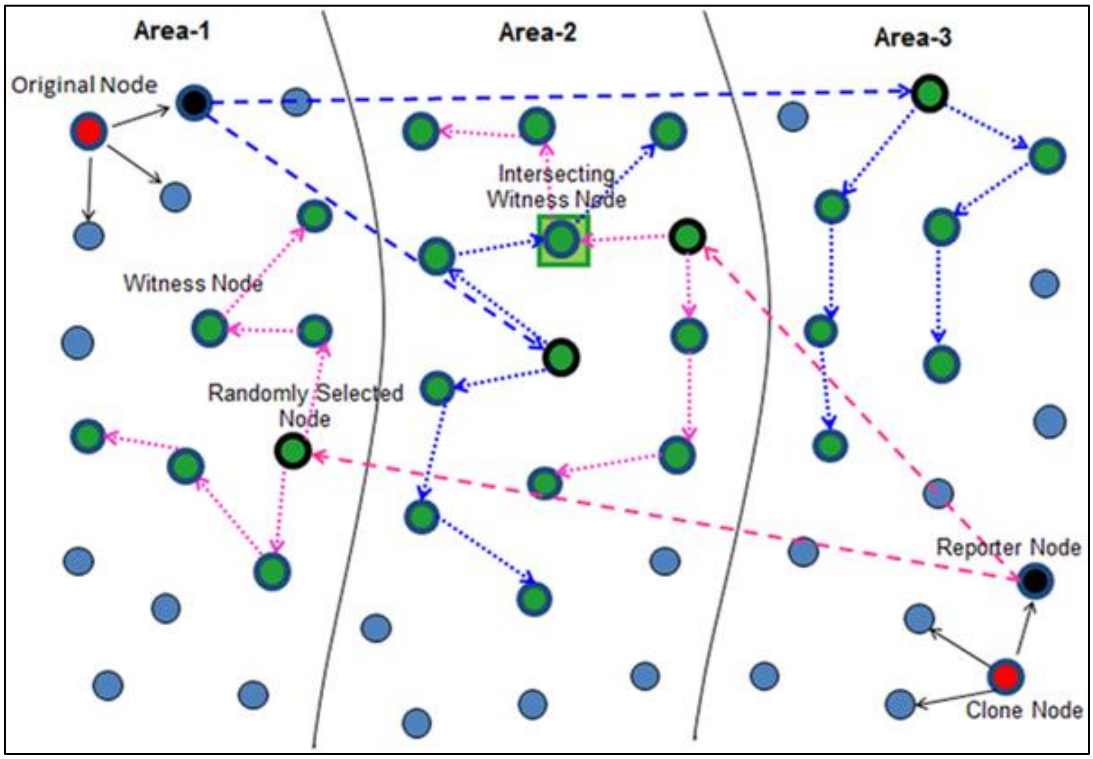


Figure 8. Improved witness node selection []

Detection of clone: When "witness node" discovers conflicting information (2 distinct "location claims with similar node ID") replicas get revoked by broadcasting the 2 clashing claims (a proof). Each hub will end link with the replicated node after they autonomously confirm the marks.

CHAPTER-4

4.1 PERFORMANCE ANALYSIS AND SIMULATION RESULTS

In previous techniques of detection adversary can very easily and safely deploy any no. of clones or replicas in the whole network. A small attack is also launched by the attacker in order to protect the nodes that it is cloning. To avoid all these attacks we must ensure the security of any WSN network. In node clone attacks if the attacker is smart then it targets to find and compromise all the witness nodes. This is possible only if the attacker comes to know from the reporter hub about the randomly chosen hub in each area from where "random walk starts". In this chapter, I have theoretically analyzed ways in which my method is better than the previously existing protocols.

4.1.1 Network Division Analysis

The network in my method is divided into various different areas. The area division depends on many factors such as:-

- Witness node security
- Size of areas
- Aggregate cost of communication
- Network Size

The minimum areas assumed in my method are 3. Because if the area is divided into 2 parts then according to the mechanism every reporter selects both the areas which makes it easy for the adversary to detect about the critical " witness nodes" . As a result it can very easily attack whole area to protect its replica or to evade detection. In case of area more than 3 the reporter is able to select area in a fully random manner as now there are many ways to select any particular area combination. This leads to better security of "witness nodes".

4.1.2 Area Selection Analysis

The zone choice system is needy upon the quantity of zones into which the entire system is isolated. In the event that the entire system is partitioned into odd number of ranges (3, 5, 7 and so on) the reporters will haphazardly choose regions and if the entire system is separated into

considerably number of zones (4, 8 et cetera) the reporters will arbitrarily choose areas. My method has lesser walk steps as compared to previously existing RAWL protocol and also it has moderate communication and memory overheads as compared to RAWL which ad high values.

4.2 SNAPSHOTS OF RESULTS

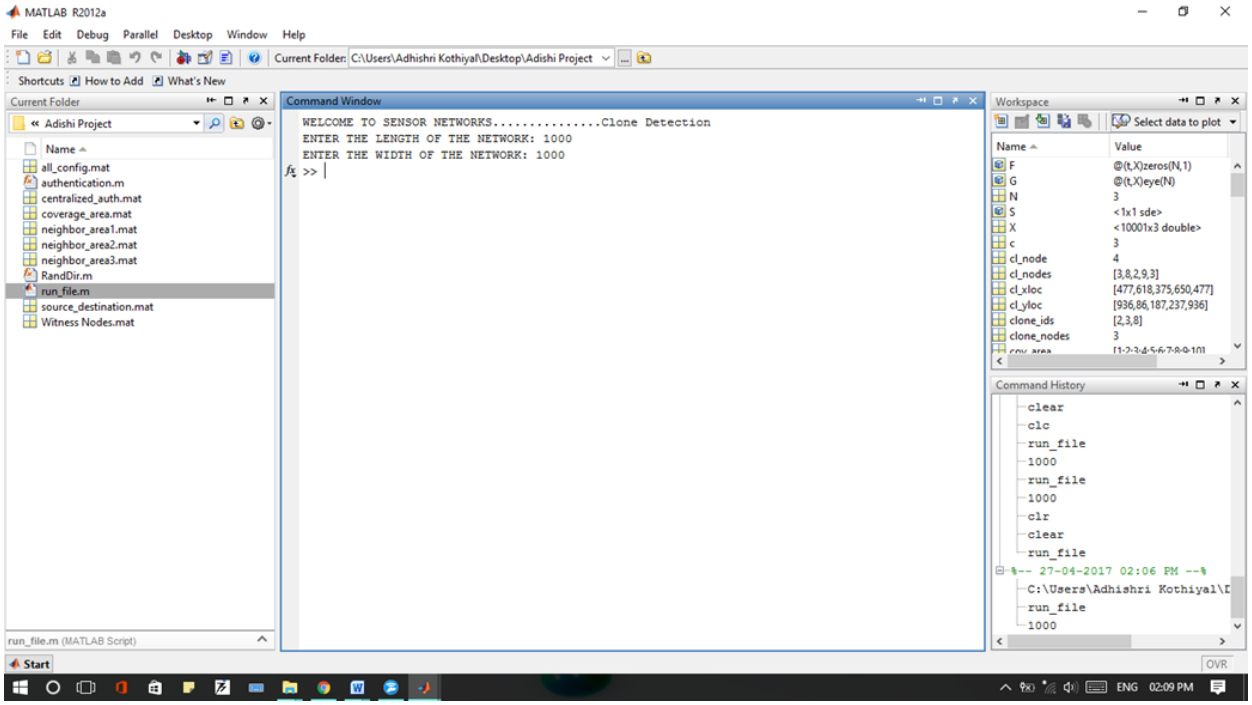


Figure 9. Network of 1000X1000 meters

When we run the program this is the first screen that displays. It basically takes the input. In my project a fixed area of 1000 as length and breath each is pre-defined in the program with 30 nodes each is defined because increasing nodes over this will increase the overheads.

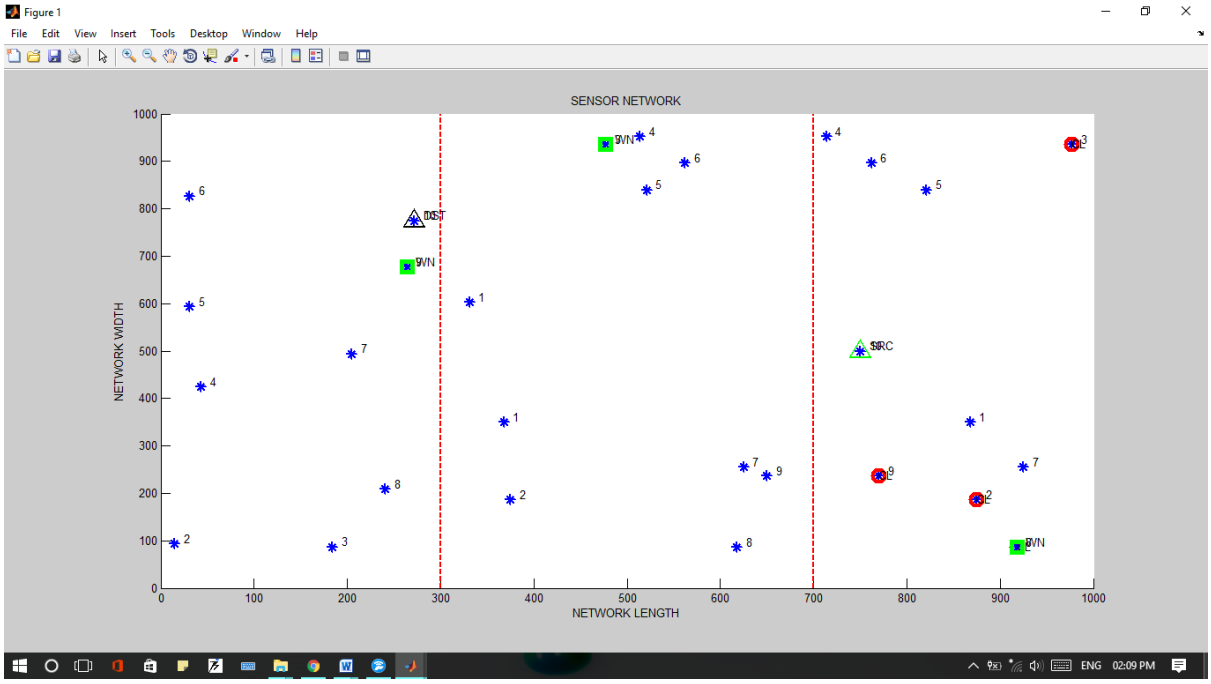


Figure10. Nodes plotted and Area Selected

This screenshot displays the 30 nodes that are plotted over 3 areas in 1000X1000 network. The red circled nodes are claimer nodes and the ones on green are witness nodes. Also it shows source and destination.

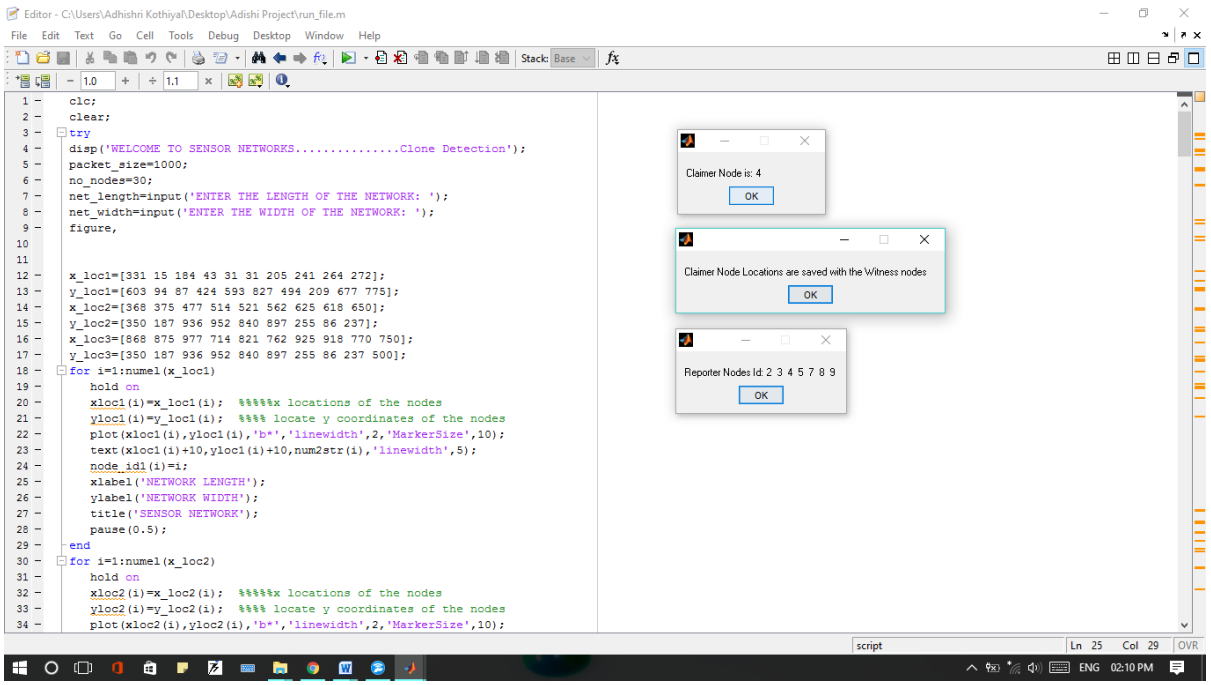


Figure11. Claimer and Reporter Node Found

Message boxes popes when claimer and reporter nodes are found. Claimer node locations are saved with witness nodes.

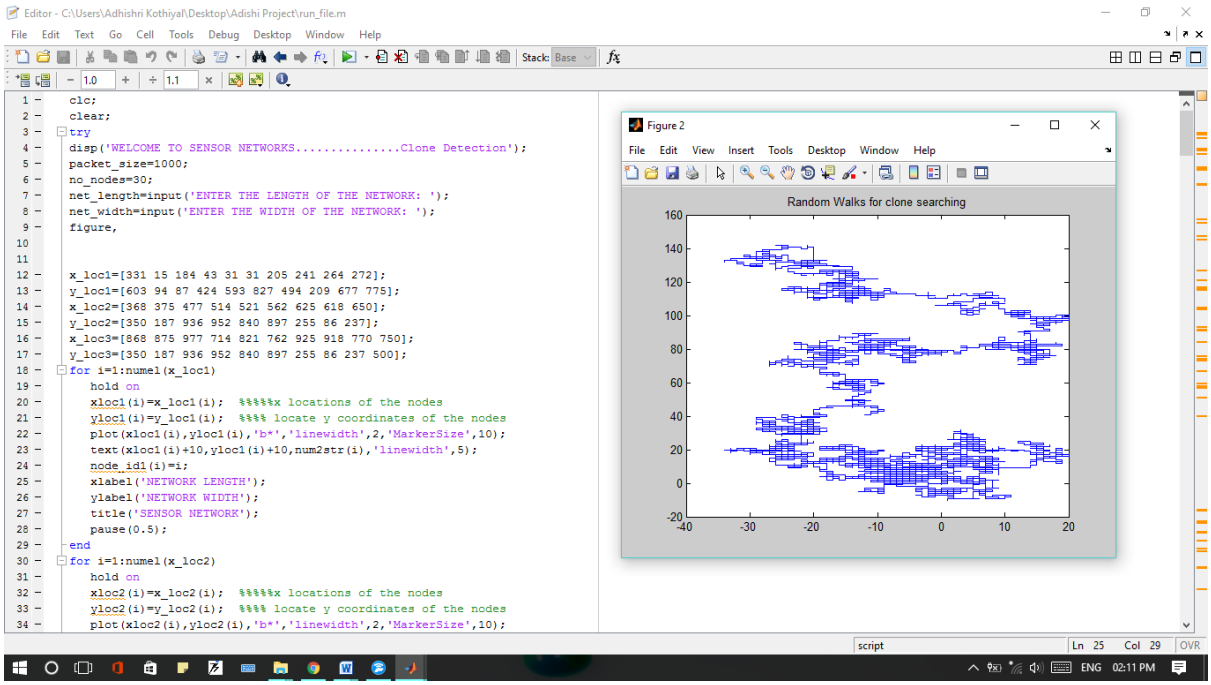


Figure12. Random Walks

Random walks being done in order to find replicas of any node all over the network.

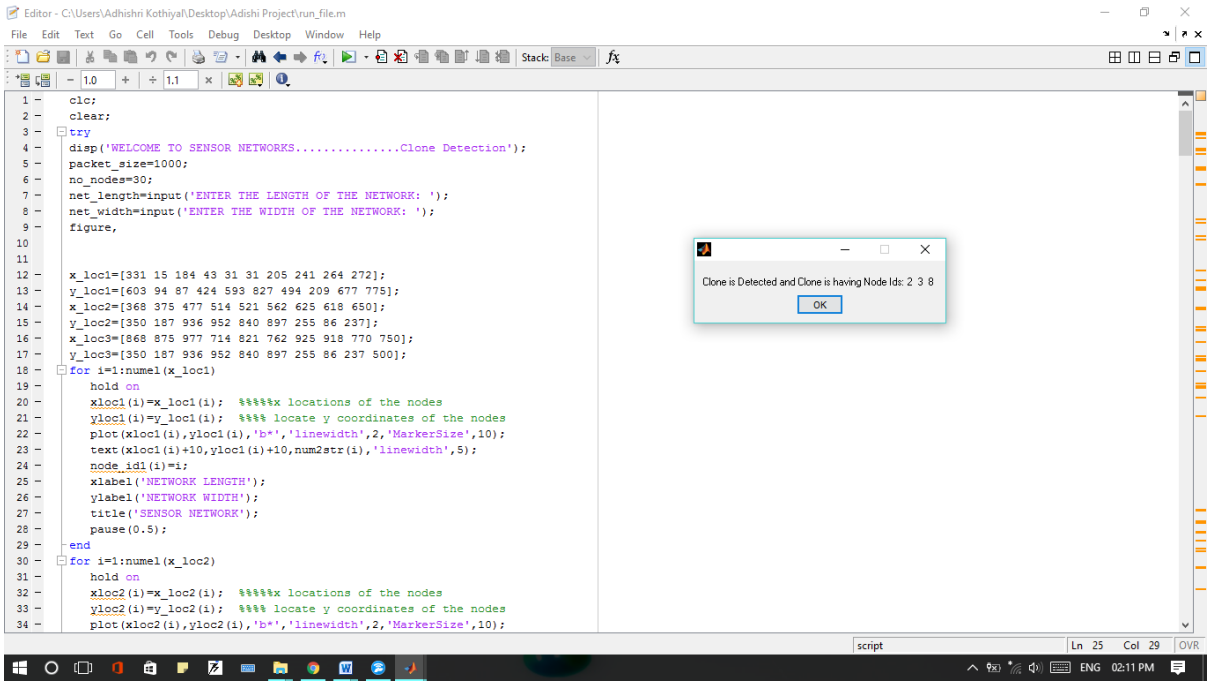


Figure13. Node Replicas Found

When any claimer node is found a message box pops up indicating node ID of the clonned nodes.

Following screenshots are of various graphs:

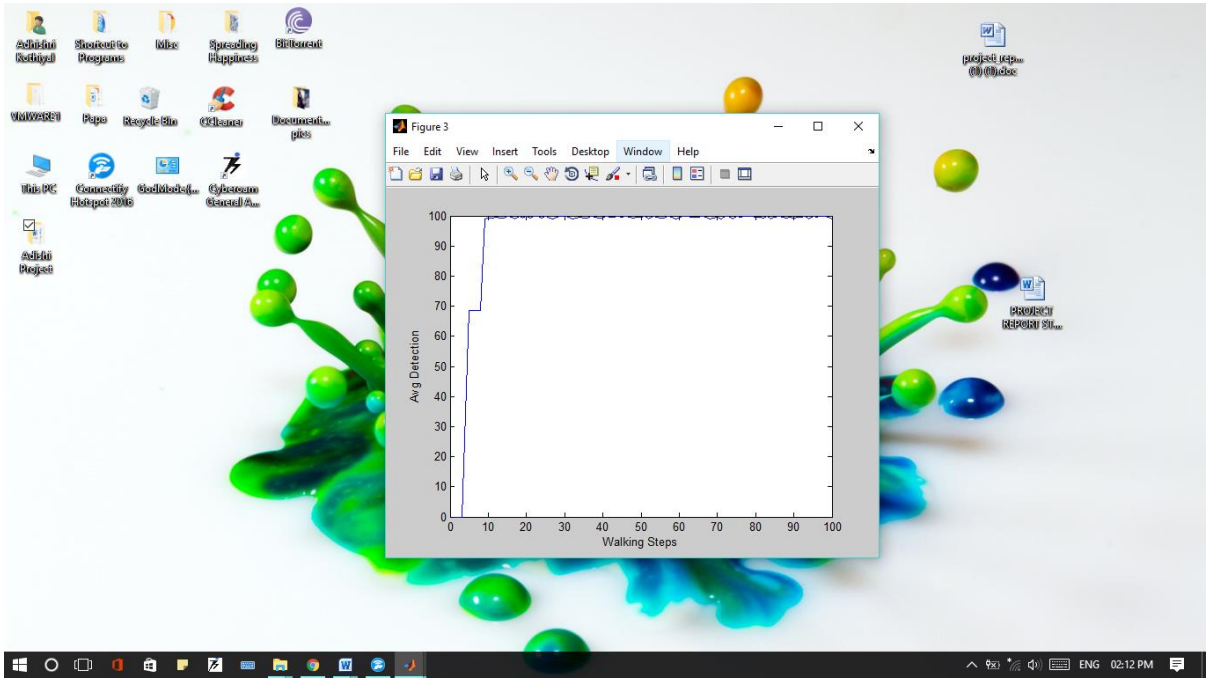


Figure14. Average Detection Vs Walking Steps

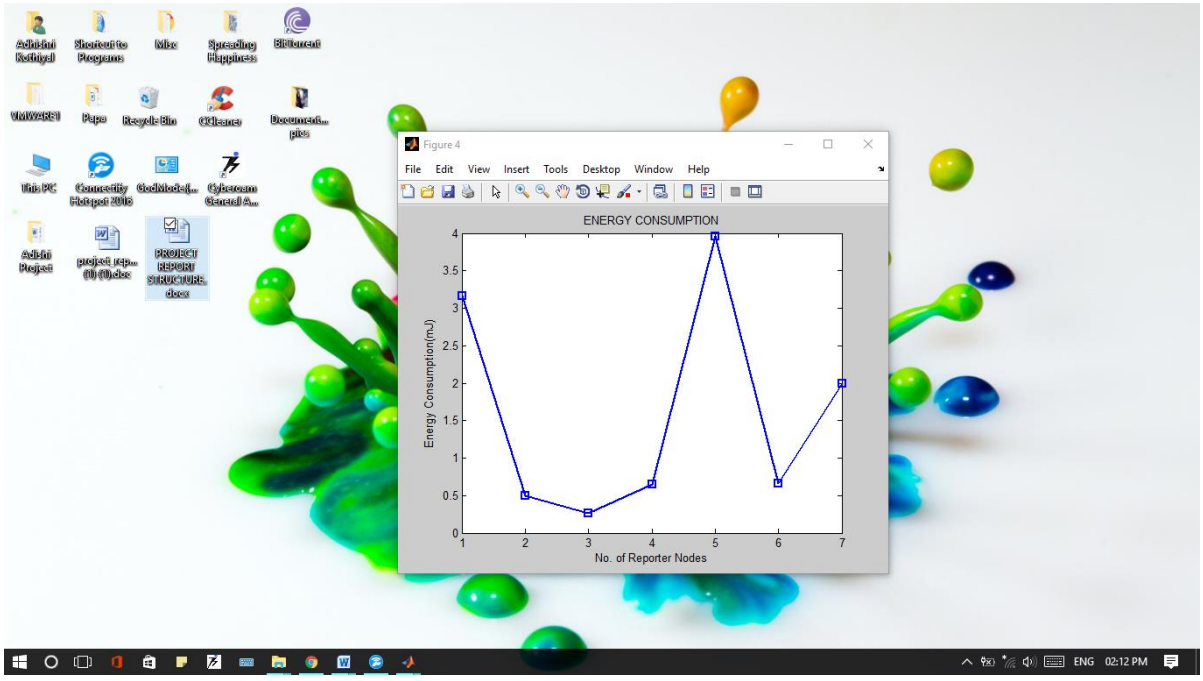


Figure15. Energy Consumption Vs Number of Reporter Nodes

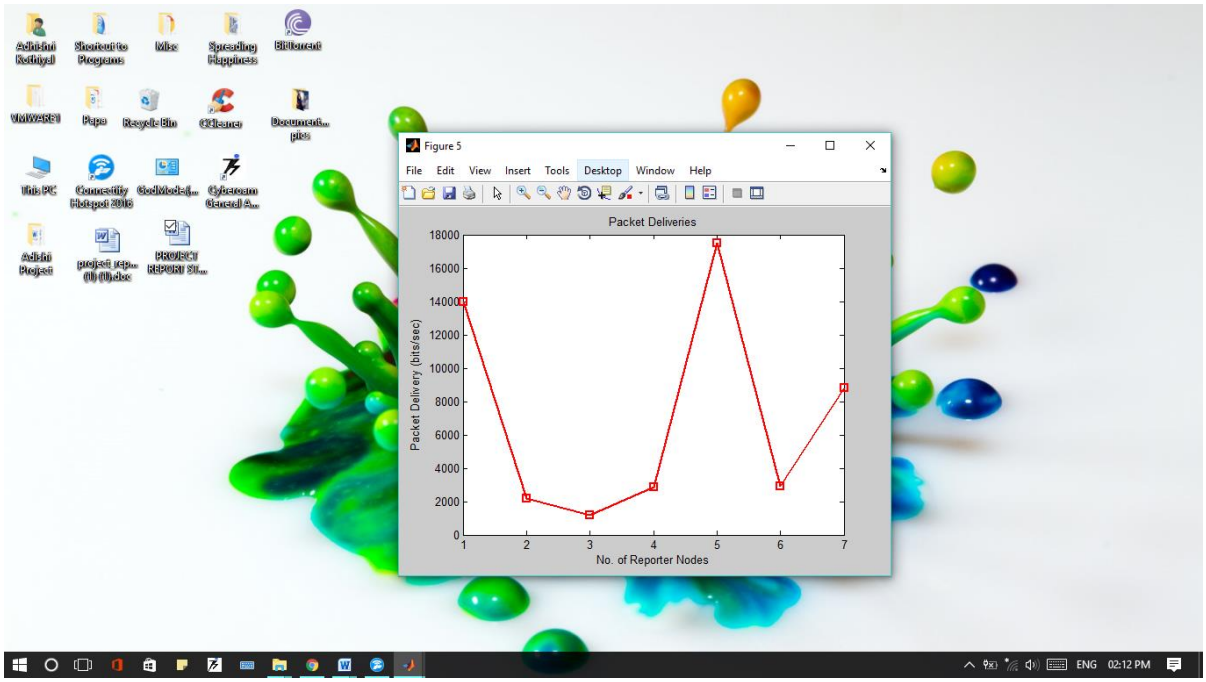


Figure16. Packet Delivery Vs Number of Reporter Nodes

CHAPTER-5

4.1 CONCLUSION

In this report, I have presented a distributed witness node based scheme called RWND for the identification of replicas in WSNs (static) which is based on the claimer-reporter-witness framework. I have brought up and clarified some huge shortcomings of the latest and promising arrangement RAWL for the identification of replicas or hub clone assaults. I have additionally decided numerous imperative inadequacies of the entire current accompanying witness hub based plans. I have enhanced the past work RWND by presenting another instrument for the witness hub choice as the witness determination which is an imperative piece of all the witness hub construct techniques in light of which the entire recognition procedure of clones is subject to. I have likewise attempted to give a hypothetical investigation of instrument of territory determination and the security examination of entire plan. The reenactment comes about contrasting new plan and RAWL. The broad reenactments affirm this new strategy beats the RAWL convention as the security of neighbor hubs is expanded essentially with fundamental correspondence, calculation and memory overheads.

REFERENCES

1. Akyildiz IF, Su W, Sankarasubramaniam Y, Cayirci E (2002). Wireless sensor networks: a survey. Computer networks, 38(4), 393–422.
2. Khan WZ, Xiang Y, Aalsalem MY, Arshad Q (2013). Mobile phone sensing systems: A survey. Communications Surveys & Tutorials, IEEE, 15(1), 402–427.
3. Zhu S, Setia S, Jajodia S, (2006). LEAP+: Efficient security mechanisms for large-scale distributed sensor networks. ACM Transactions on Sensor Networks (TOSN), 2(4), 500–528.
4. Karlof, C, Sastry N, Wagner D, (2004). TinySec: a link layer security architecture for wireless sensor networks. In Proceedings of the 2nd international conference on Embedded networked sensor systems (pp. 162–175). ACM.
5. Perrig A, Szewczyk R, Tygar J, Wen V, Culler D (2002) SPINS: security protocols for sensor networks. Wireless Networks, 2002; 8: 521–34.
6. Hartung C, Balasalle J, Han R (2005). Node compromise in sensor networks: The need for secure systems. Department of Computer Science University of Colorado at Boulder.
7. Karlof C, Wagner D (2003). Secure routing in wireless sensor networks: Attacks and countermeasures. Ad hoc networks, 1(2), 293–315
8. Wood A, Stankovic J (2002) Denial of Service in Sensor Networks. IEEE Computer. 2002 October; 3(10):54–62.
9. Choi H, Zhu S, La Porta, TF (2007). SET: Detecting node clones in sensor networks. In Security and Privacy in Communications Networks and the Workshops, 2007. SecureComm 2007. Third International Conference on (pp. 341–350). IEEE.
10. Brooks R, Govindaraju PY, Pirretti M, Vijaykrishnan N, Kandemir MT (2007). On the detection of clones in sensor networks using random key predistribution. Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on, 37(6), 1246–1258.
11. Eschenauer L, Gligor, VD (2002). A key-management scheme for distributed sensor networks. In Proceedings of the 9th ACM conference on Computer and communications security (pp. 41–47). ACM.
12. Xing K, Liu F, Cheng X, Du DHC (2008). Real-time detection of clone attacks in wireless sensor networks. In Distributed Computing Systems, 2008. ICDCS'08. The 28th International Conference on (pp. 3–10). IEEE.

13. Xing K, Cheng X, Ma L, Liang Q (2007). Superimposed code based channel assignment in multi-radio multi-channel wireless mesh networks. In Proceedings of the 13th annual ACM international conference on Mobile computing and networking (pp. 15–26). ACM.
14. Znaidi W, Minier M, Ubéda S (2009). Hierarchical node replication attacks detection in wireless sensors networks. In Personal, Indoor and Mobile Radio Communications, 2009 IEEE 20th International Symposium on (pp. 82–86). IEEE.
15. Yu CM, Lu CS, Kuo, SY (2012). CSI: compressed sensing-based clone identification in sensor networks. In Pervasive Computing and Communications Workshops (PERCOM Workshops), 2012 IEEE International Conference on (pp. 290–295). IEEE.
16. Conti M, Di Pietro R, Mancini LV, Mei A (2007). A randomized, efficient, and distributed protocol for the detection of node replication attacks in wireless sensor networks. In Proceedings of the 8th ACM international symposium on Mobile ad hoc networking and computing (pp. 80–89). ACM.
17. Conti M, Di Pietro R, Mancini LV, Mei A (2011). Distributed detection of clone attacks in wireless sensor networks. Dependable and Secure Computing, IEEE Transactions on, 8(5), 685–698.
18. Bekara C, Laurent-Maknavicius M (2007). A new protocol for securing wireless sensor networks against nodes replication attacks. In Wireless and Mobile Computing, Networking and Communications, 2007. WiMOB 2007. Third IEEE International Conference on (pp. 59–59). IEEE.
19. Bekara C, Laurent-Maknavicius M (2012). Defending against nodes replication attacks on wireless sensor networks.
20. Ko LC, Chen HY, Lin GR (2009). A neighbor-based detection scheme for wireless sensor networks against node replication attacks. In Ultra Modern Telecommunications & Workshops, 2009. ICUMT'09. International Conference on (pp. 1–6). IEEE.
21. Ho JW (2010). Distributed detection of node capture attacks in wireless sensor networks. INTECH Open Access Publisher.
22. Ho JW, Liu D, Wright M, Das SK (2009). Distributed detection of replica node attacks with group deployment knowledge in wireless sensor networks. Ad Hoc Networks, 7(8), 1476–1488.

23. Sei Y, Honiden S (2008). Distributed detection of node replication attacks resilient to many compromised nodes in wireless sensor networks. In Proceedings of the 4th Annual International Conference on Wireless Internet (p. 28). ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).
24. Parno B, Perrig A, Gligor V (2005). Distributed detection of node replication attacks in sensor networks. In Security and Privacy, 2005 IEEE Symposium on (pp. 49–63). IEEE.
25. Zhu B, Addada VGK, Setia S, Jajodia S, Roy S (2007). Efficient distributed detection of node replication attacks in sensor networks. In Computer Security Applications Conference, 2007. ACSAC 2007. Twenty-Third Annual (pp. 257–267). IEEE.
26. Zhu B, Setia S, Jajodia S, Roy S, Wang L (2010). Localized multicast: efficient and distributed replica detection in large-scale sensor networks. Mobile Computing, IEEE Transactions on, 9(7), 913–926.
27. Zhang M, Khanapure V, Chen S, Xiao X (2009). Memory efficient protocols for detecting node replication attacks in wireless sensor networks. In Network Protocols, 2009. ICNP 2009. 17th IEEE International Conference on (pp. 284–293). IEEE.
28. Zeng Y, Cao J, Zhang S, Guo S, Xie L (2010). Random-walk based approach to detect clone attacks in wireless sensor networks. Selected Areas in Communications, IEEE Journal on, 28(5), 677–691.
29. Khan WZ, Aalsalem MY, Saad NM, Xaing Y, Luan TH (2014). Detecting replicated nodes in Wireless Sensor Networks using random walks and network division. In Wireless Communications and Networking Conference (WCNC), 2014 IEEE (pp. 2623–2628). IEEE.
30. Melchor C, Ait-Salem B, Gaborit P, Tamine K (2009) Active detection of node replication attacks. International Journal of Computer Science and Network Security, 2009; 9(2):13–21.
31. Li Z, Gong G (2009). Randomly directed exploration: An efficient node clone detection protocol in wireless sensor networks. In Mobile Adhoc and Sensor Systems, 2009. MASS'09. IEEE 6th International Conference on (pp. 1030–1035). IEEE.
32. Chano KIM, Seungjae SHIN, Chanil PARK (2009). A resilient and efficient replication attack detection scheme for wireless sensor networks. IEICE transactions on information and systems, 92(7), 1479–1483.
33. Zhu C, Sun S, Wang L, Ding S, Wang J, Xia C (2014) Promotion of cooperation due to diversity of players in the spatial public goods game with increasing neighborhood

size. Physica A: Statistical Mechanics and its Applications 406, 145–154. Online publication date: 1-Jul-2014.

34. Xia C, Miao Q, Wang J, Ding S (2014) Evolution of cooperation in the traveler's dilemma game on two coupled lattices. Applied Mathematics and Computation 246, 389–398. Online publication date: 1-Nov-2014.
35. Wang L, Li X, Zhang YQ, Zhang Y, Zhang K (2011). Evolution of scaling emergence in large-scale spatial epidemic spreading. PloS one, 6(7), e21197. doi: 10.1371/journal.pone.0021197. pmid:21747932
36. Zhang Y, Wang L, Zhang YQ, Li X (2012). Towards a temporal network analysis of interactive WiFi users. EPL (Europhysics Letters), 98(6), 68002.
37. Wang L, Li X (2014). Spatial epidemiology of networked metapopulation: An overview. Chinese Science Bulletin, 59(28), 3511–3522.
38. Wang L, Wang Z, Zhang Y Li, X (2013). How human location-specific contact patterns impact spatial transmission between populations?. Scientific reports, 3.
39. Dutertre B, Cheung S, Levy J (2004). Lightweight key management in wireless sensor networks by leveraging initial trust. Technical Report SRI-SDL-04-02, SRI International.
40. Ratnasamy S, Karp B, Yin L, Yu F, Estrin D, Govindan R, et al. (2002). GHT: a geographic hash table for data-centric storage. In Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications (pp. 78–87). ACM.
41. Khan WZ, Aalsalem MY, Saad MNBM, Xiang Y (2013). Detection and mitigation of node replication attacks in wireless sensor networks: a survey. International Journal of Distributed Sensor Networks, 2013.
42. Zhu WT, Zhou J, Deng RH, Bao F (2012). Detecting node replication attacks in wireless sensor networks: a survey. Journal of Network and Computer Applications, 35(3), 1022–1034.
43. Khan WZ, Saad MNBM, Aalsalem M. Y (2013). Scrutinising well-known countermeasures against clone node attack in mobile wireless sensor networks. International Journal of Grid and Utility Computing, 4(2), 119–127.
44. Seshadri A, Perrig A, Van Doorn L, Khosla P (2004). Swatt: Software-based attestation for embedded devices. In Security and Privacy, 2004. Proceedings. 2004 IEEE Symposium on (pp. 272–282). IEEE.

45. Aalsalem MY, Taheri J, Zomaya AY (2010). A framework for real time communication in sensor networks. In Computer Systems and Applications (AICCSA), 2010 IEEE/ACS International Conference on (pp. 1–7). IEEE.
46. Karp B, Kung HT (2000). GPSR: Greedy perimeter stateless routing for wireless networks. In Proceedings of the 6th annual international conference on Mobile computing and networking (pp. 243–254). ACM.
47. Shah R. C, Roy S, Jain S, Brunette W (2003). Data mules: Modeling and analysis of a three-tier architecture for sparse sensor networks. Ad Hoc Networks, 1(2), 215–233.
48. Aldous D, Fill J (2002). Reversible Markov chains and random walks on graphs. [Online]. Available: Ellis R (2001). Torus hitting times from green's functions.
49. Menezes AJ, Van Oorschot PC, Vanstone SA (1996). Handbook of applied cryptography. CRC press.
50. Zuniga M, Avin C, Hauswirth M (2010). Querying dynamic wireless sensor networks with non-revisiting random walks. In Wireless Sensor Networks (pp. 49–64). Springer Berlin Heidelberg.
51. Friedman R, Kliot G, Avin C (2010). Probabilistic quorum systems in wireless ad hoc networks. ACM Transactions on Computer Systems (TOCS), 28(3), 7.
52. Estrin D, Govindan R, Heidemann J, Kumar S (1999). Next century challenges: Scalable coordination in sensor networks. In Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking (pp. 263–270). ACM.
53. Liu A, Ning P (2008). TinyECC: A configurable library for elliptic curve cryptography in wireless sensor networks. In Information Processing in Sensor Networks, 2008. IPSN'08. International Conference on (pp. 245–256). IEEE.
54. www.google.com/images

APPENDICES

CODE:

```
clc;
clear;
try
disp('WELCOME TO SENSOR NETWORKS..............Clone Detection');
packet_size=1000;
no_nodes=30;
net_length=input('ENTER THE LENGTH OF THE NETWORK: ');
net_width=input('ENTER THE WIDTH OF THE NETWORK: ');
figure,



x_loc1=[331 15 184 43 31 31 205 241 264 272];
y_loc1=[603 94 87 424 593 827 494 209 677 775];
x_loc2=[368 375 477 514 521 562 625 618 650];
y_loc2=[350 187 936 952 840 897 255 86 237];
x_loc3=[868 875 977 714 821 762 925 918 770 750];
y_loc3=[350 187 936 952 840 897 255 86 237 500];
for i=1:numel(x_loc1)
   hold on
   xloc1(i)=x_loc1(i);  %%%%%x locations of the nodes
   yloc1(i)=y_loc1(i);  %%%% locate y coordinates of the nodes
   plot(xloc1(i),yloc1(i),'b*','linewidth',2,'MarkerSize',10);
   text(xloc1(i)+10,yloc1(i)+10,num2str(i),'linewidth',5);
   node_id1(i)=i;
   xlabel('NETWORK LENGTH');
   ylabel('NETWORK WIDTH');
   title('SENSOR NETWORK');
   pause(0.5);
end
for i=1:numel(x_loc2)
   hold on
   xloc2(i)=x_loc2(i);  %%%%%x locations of the nodes
   yloc2(i)=y_loc2(i);  %%%% locate y coordinates of the nodes
   plot(xloc2(i),yloc2(i),'b*','linewidth',2,'MarkerSize',10);
   text(xloc2(i)+10,yloc2(i)+10,num2str(i),'linewidth',5);
   node_id2(i)=i;
   xlabel('NETWORK LENGTH');
   ylabel('NETWORK WIDTH');
```

```
   title('SENSOR NETWORK');
   pause(0.5);
end


for i=1:numel(x_loc3)
   hold on
   xloc3(i)=x_loc3(i);  %%%%%x locations of the nodes
   yloc3(i)=y_loc3(i);  %%%% locate y coordinates of the nodes
   node_id3(i)=i;
   plot(xloc3(i),yloc3(i),'b*','linewidth',2,'MarkerSize',10);
   text(xloc3(i)+10,yloc3(i)+10,num2str(i),'linewidth',5);
   xlabel('NETWORK LENGTH');
   ylabel('NETWORK WIDTH');
   title('SENSOR NETWORK');
   pause(0.5);
end


for jp=1:no_nodes
   energy_nodes(jp)=no_nodes*rand;
   ini_delay(jp)=rand;
end

x1=[300 300];
y1=[0 1000];
line(x1,y1,'Color','r','LineStyle','--','linewidth',2);
hold on;
x2=[700 700];
y2=[0 1000];
line(x2,y2,'Color','r','LineStyle','--','linewidth',2);


source=round(no_nodes*rand);
hold on
if source==0
   source=12;
end
```

```
dest=round(no_nodes*rand);
if dest==0
   dest=9;
end



figure(1),
if source>=0 & source<=10

plot(xloc1(numel(x_loc1*rand)),yloc1(numel(x_loc1*rand)),'g^','linewidth',2
,'MarkerSize',15);
    text(xloc1(numel(y_loc1*rand))+10,yloc1(numel(y_loc1*rand))+10,'SRC');
elseif source>=11 & source<=19

plot(xloc2(numel(x_loc2*rand)),yloc2(numel(x_loc2*rand)),'g^','linewidth',2
,'MarkerSize',15);
    text(xloc2(numel(y_loc2*rand))+10,yloc2(numel(y_loc2*rand))+10,'SRC');
else

plot(xloc3(numel(x_loc3*rand)),yloc3(numel(x_loc3*rand)),'g^','linewidth',2
,'MarkerSize',15);
    text(xloc3(numel(x_loc3*rand))+10,yloc3(numel(x_loc3*rand))+10,'SRC');
end



hold on;
if dest>=0 & dest<=10

plot(xloc1(numel(x_loc1*rand)),yloc1(numel(x_loc1*rand)),'k^','linewidth',2
,'MarkerSize',15);
    text(xloc1(numel(y_loc1*rand))+10,yloc1(numel(y_loc1*rand))+10,'DST');
elseif dest>=11 & dest<=19

plot(xloc2(numel(x_loc2*rand)),yloc2(numel(x_loc2*rand)),'k^','linewidth',2
,'MarkerSize',15);
    text(xloc2(numel(y_loc2*rand))+10,yloc2(numel(y_loc2*rand))+10,'DST');
else
```

```
plot(xloc3(numel(x_loc3*rand)),yloc3(numel(x_loc3*rand)),'^k','linewidth',2
,'MarkerSize',15);
    text(xloc3(numel(x_loc3*rand))+10,yloc3(numel(x_loc3*rand))+10,'DST');
end




% Clone Nodes %

x_clones=[];
y_clones=[];



   dp=20*rand;
   if dp<5
      for x=1:5
          clone_nodes=round(numel(xloc1)*rand);
          if clone_nodes==0
              clone_nodes=4;
          end
          cl_nodes(x)=clone_nodes;
      end
      figure(1)

plot(xloc1(cl_nodes),yloc1(cl_nodes),'ro','linewidth',3,'MarkerSize',11);
      text(xloc1(cl_nodes),yloc1(cl_nodes),'CL');


   end
   if dp>5 & dp<=10
          for x=1:5
              clone_nodes=round(numel(xloc2)*rand);
              if clone_nodes==0
                  clone_nodes=5;
              end
              cl_nodes(x)=clone_nodes;
              cl_xloc(x)=xloc1(clone_nodes);
              cl_yloc(x)=yloc1(clone_nodes);
```

```
            end
            figure(1)

plot(xloc2(cl_nodes),yloc2(cl_nodes),'ro','linewidth',3,'MarkerSize',11);
            text(xloc2(cl_nodes),yloc2(cl_nodes),'CL');
   end
   if dp>10
            for x=1:5
               clone_nodes=round(numel(xloc3)*rand);
               if clone_nodes==0
                   clone_nodes=7;
               end
               cl_nodes(x)=clone_nodes;
               cl_xloc(x)=xloc2(clone_nodes);
               cl_yloc(x)=yloc2(clone_nodes);
            end
            figure(1)

plot(xloc3(cl_nodes),yloc3(cl_nodes),'ro','linewidth',3,'MarkerSize',11);
            text(xloc3(cl_nodes),yloc3(cl_nodes),'CL');

   end



% For Cover Area 1 %
for i=1:numel(x_loc1)
    for j=2:numel(x_loc1)
       dist_c=sqrt((xloc1(i)-xloc1(j))^2+(yloc1(i)-yloc1(j))^2);
       cov_area(i,1)=node_id1(i);
       if dist_c<400
                cov_area1(i,j)=node_id1(j);
                dist_area1(i,j)=dist_c;
       end
    end

end

save('neighbor_area1','cov_area1','dist_area1');
[rs1,cs1]=size(cov_area1);
```

```
for i=1:rs1
   dist_nod1(i)=sum(dist_area1(i,:));
end

[min_dist1 id1]=sort(dist_nod1,'descend');



% For Cover Area 2 %
for i=1:numel(x_loc2)
    for j=2:numel(x_loc2)
       dist_c=sqrt((xloc2(i)-xloc2(j))^2+(yloc2(i)-yloc2(j))^2);
       cov_area(i,1)=node_id2(i);
       if dist_c<400
                cov_area2(i,j)=node_id2(j);
                dist_area2(i,j)=dist_c;
       end
    end

end
save('neighbor_area2','cov_area2','dist_area2');
[rs2,cs2]=size(cov_area2);
for i=1:rs2
   dist_nod2(i)=sum(dist_area2(i,:));
end

[min_dist2 id2]=sort(dist_nod2,'descend');



% For Cover Area 3 %
for i=1:numel(x_loc3)
    for j=2:numel(x_loc3)
       dist_c=sqrt((xloc3(i)-xloc3(j))^2+(yloc3(i)-yloc3(j))^2);
       cov_area(i,1)=node_id3(i);
       if dist_c<400
                cov_area3(i,j)=node_id3(j);
                dist_area3(i,j)=dist_c;
       end
    end
```

```
end
save('neighbor_area3','cov_area3','dist_area3');
[rs3,cs3]=size(cov_area3);
for i=1:rs3
   dist_nod3(i)=sum(dist_area3(i,:));
end

[min_dist3 id3]=sort(dist_nod3,'descend');

save all_config

load all_config
[ nodes_authent ] = authentication( no_nodes );

cl_node=id1(1);
msgbox(['Claimer Node is: ',num2str(cl_node)]);   % Claimer node id

reporter_node=cov_area1(cl_node,:);   % Reporter nodes ids
reporter_node(find(reporter_node==0))=[];

pause(1);
msgbox(['Reporter Nodes Id: ',num2str(reporter_node)]);
prob_rep_node=rand(1,numel(reporter_node));    % Probabilities of reporter
nodes



pause(1);
%Area selection%
no_areas=3;
sel_area=round(no_areas*rand);
if sel_area==0
   sel_area=2;
end



% Selecting witness node in each area %
wit_nodes1=[];
wit_nodes1(1,1)=round(numel(xloc1)*rand);
```

```
wit_nodes1(1,2)=xloc1(cl_node(1));
wit_nodes1(1,3)=yloc1(cl_node(1));

wit_nodes2=[];
wit_nodes2(1,1)=round(numel(xloc2)*rand);
wit_nodes2(1,2)=xloc1(cl_node(1));
wit_nodes2(1,3)=yloc1(cl_node(1));

wit_nodes3=[];
wit_nodes3(1,1)=round(numel(xloc3)*rand);
wit_nodes3(1,2)=xloc1(cl_node(1));
wit_nodes3(1,3)=yloc1(cl_node(1));

save('Witness Nodes','wit_nodes1','wit_nodes2','wit_nodes3');
msgbox('Claimer Node Locations are saved with the Witness nodes');

pause(1);

figure(1);
plot(xloc1(wit_nodes1(1)),yloc1(wit_nodes1(1)),'gs','linewidth',4,'MarkerSi
ze',11);
text(xloc1(wit_nodes1(1))+10,yloc1(wit_nodes1(1))+10,'WN');
hold on;
plot(xloc2(wit_nodes2(1)),yloc2(wit_nodes2(1)),'gs','linewidth',4,'MarkerSi
ze',11);
text(xloc2(wit_nodes2(1))+10,yloc2(wit_nodes2(1))+10,'WN');
hold on;
plot(xloc3(wit_nodes3(1)),yloc3(wit_nodes3(1)),'gs','linewidth',4,'MarkerSi
ze',11);
text(xloc3(wit_nodes3(1))+10,yloc3(wit_nodes3(1))+10,'WN');

% Random Walks %

N = no_areas;
F = @(t,X) zeros(N,1);
G = @(t,X) eye(N);
S = sde(F,G,'startState',zeros(N,1));

X = S.simByEuler(10000,'ntrials',1,'Z',@(t,X) RandDir(N));
```

```
pause(1);
figure,
axis off;
grid on;
comet(X(:,1),X(:,2));
title('Random Walks for clone searching');
plot(X(:,1),X(:,2));
title('Random Walks for clone searching');


% Finding Conflicts %

p=1;
q=1;
r=1;
for i=1:numel(xloc1)
   if xloc1(i)==cl_xloc(1) | xloc1(i)==cl_xloc(2) | xloc1(i)==cl_xloc(3)
      clone_ids(p)=i;
      p=p+1;



   end
end

for i=1:numel(xloc2)
   if xloc2(i)==cl_xloc(1) | xloc2(i)==cl_xloc(2) | xloc2(i)==cl_xloc(3)
      clone_ids(q)=i;
      q=q+1;

   end
end

for i=1:numel(xloc3)
   if xloc3(i)==cl_xloc(1) | xloc3(i)==cl_xloc(2) | xloc3(i)==cl_xloc(3)
      clone_ids(r)=i;
      r=r+1;
   end
```

```
end

clone_ids(find(clone_ids==0))=[];
msgbox(['Clone is Detected and Clone is having Node Ids:
',num2str(clone_ids)]);

[r,c]=size(X);
max_prob=100;
for i=1:r

det_probability(i)=abs(X(i,1)*numel(clone_ids)*sum(min_dist1))/packet_size;
    if det_probability(i)>max_prob
        det_probability(i)=abs(max_prob-rand);
    end

end

pause(1);
figure,
plot(det_probability(1:100));
xlabel('Walking Steps');
ylabel('Avg Detection');
total_min_dist=(min_dist1(1)+min_dist2(1)+min_dist3(1));
for i=1:numel(reporter_node)
    energy_consump(i)=(energy_nodes(i)*ini_delay(i))/numel(reporter_node);
    packet_del(i)=energy_consump(i).*total_min_dist;
end

figure,
plot(energy_consump,'-bs','linewidth',2);
xlabel('No. of Reporter Nodes');
ylabel('Energy Consumption(mJ)');
title('ENERGY CONSUMPTION');

pause(1);
figure,
plot(packet_del,'-rs','linewidth',2);
xlabel('No. of Reporter Nodes');
ylabel('Packet Delivery (bits/sec)');
title('Packet Deliveries');
```

```
catch
    msgbox('Something Went wrong..........Kindly Run again.......');
end
```