

REPLICATION OF ATTACKS IN A WIRELESS SENSOR NETWORK USING NS2

Tejaswi Singh¹, Aatish Gandotra²

¹Student, Computer Science Engineering, Guru Tegh Bahadur Institute of Technology, Delhi, India

²Student, Electronics and Communication Engineering, Guru Tegh Bahadur Institute of Technology, Delhi, India

Abstract

A Wireless Sensor Network (WSN) comprises of sovereign sensor devices that are used to supervise physical and environmental conditions like temperature and pressure. The WSN is built of hundreds and thousands of recognizing stations called nodes, where each node consists of one or more sensors having a radio transceiver, an internal/external antenna, a microcontroller and a battery. Wireless sensor networks are the systems that are used to communicate by sensing the behavioral changes and the sensing nodes will collect the data and it will get handled. After data handling, the data will be sent to the receiver.

The wireless sensor networks have to be fortified from network attacks especially at unfavorable situations because data can easily be obtained by the attackers. There are also some security protocols being implemented in sensor networks. There are some limitations in a wireless sensor network like they have limited storage capacity, limited capability of processing and limited energy to transmit data. These drawbacks can make wireless sensor network different from other networks. The imitation of the attacks are done in the NS2 simulator. By imitating, the performance of the network can be monitored.

Keywords: Network Security, Wireless, Sensor, Internet, System Security, Simulator, NS2, Simulation of attacks.

1. INTRODUCTION TO WSN

Figure 1.1 represents the architecture of a WSN. The WSN is built of hundreds and thousands of detection stations called nodes, where each node connects to sensors

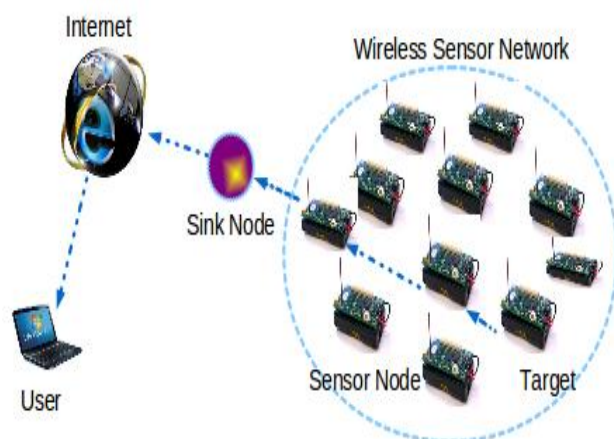


Fig 1.1: Illustration of a WSN

Constructing a wireless sensor network (WSN) has become important in all places. The sensor nodes collect the data and direct it to the center station for processing and then it is directed to the user via a wireless medium. A WSN has copious applications in many fields. They are employed in many places. A WSN is used in these applications to supervise the safeguarding, improving the throughput and enhance the defense and safety. For wide deployment, it is

required that the sensors should be made smaller and easy on the pocket. There are also multifarious methods being proposed to safeguard the network from different manners of attacks. Figure 1.2 shows the applications of WSN's in copious fields. They are employed in many places and the sensors have a capability to give warning in tragic situations.

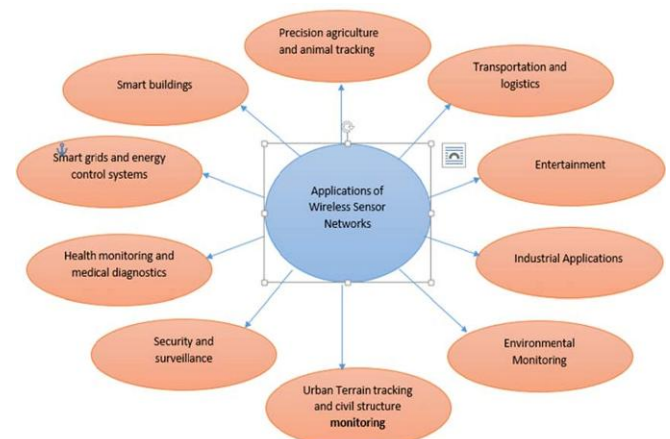


Fig 1.2: Applications of a WSN

1.1 Existing Systems

There are certain systems proposed earlier that deal with providing secure data transfer in a wireless sensor network, but they have their own disadvantages. Present wireless sensor networks have restricted functionalities, so illicit users can easily access the WSN and they can easily change data integrity, introduce a wrong message and destroy the

network without any user association. [1] For the purpose of providing a secure data communication over the network, several cryptographic and other techniques have been implemented. A WSN is more susceptible to several attacks/threats. [4] In spite of attaining public key cryptography, the security level is not proper in WSN. The data was encrypted and decrypted at the receivers end. The attackers could even attack the encrypted data. In the existing systems, the security of the data integrity is fragile. The attacker can attack all the information in the transitional nodes which violates data confidentiality.

1.2 Introduction to NS2 Simulator

Using the network simulator NS2, the attacks in the WSN can be replicated. NS2 makes a replica of a real time network. It is a time based event driven simulator. The code can be written in such a way that at a particular time, what particular event can happen. The data transfer between the nodes and the attacks can be shown. It has become one of the most widely used open source simulators. It is a free simulation tool that can be available online [14] [19]. The simulator consists of a wide variety of applications, protocols like TCP, UDP and many network parameters. It runs on various platforms like UNIX, Mac and windows platforms. This NS2 tool allows to develop a model design for wireless sensor network connection between nodes in the network. Based on the network attacks like denial of service [4], hello flood attack, sinkhole attacks, Sybil attack the network security can be tested. These attacks can be created in the network and the security level of the wireless sensor network can be tested to ensure secure data transmission between the nodes in the network.

Figure 1.3 shows the basic architecture of NS2 Simulator [13]. It is provided with a command 'ns' to execute the code written in NS2. The name of the Tcl simulation script is passed as an input argument. After executing a simulation trace file is created which can be used to create animation or to plot graph. NS2 Simulator consists of two languages namely C++ and OTcl (Object oriented Tool Command Language). C++ does the internal mechanism i.e. back end and OTcl deals with the front end [12].

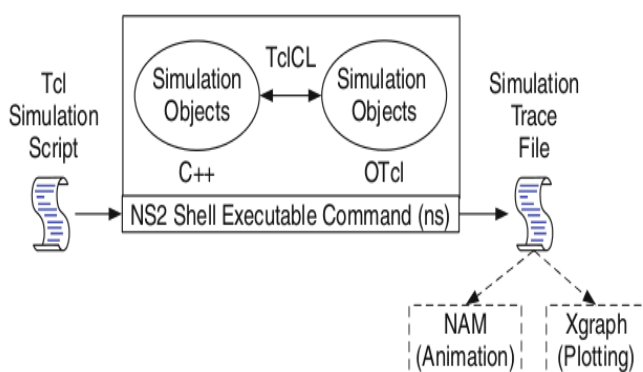


Fig 1.3: Basic Architecture of a NS2 Simulator [13]

The simulation trace file engendered after execution can be used to create animation in a network animator or to plot a graph. The information in the network animator can be logged in data format in namtrace file.

There are many advantages of NS2 Simulator:

- It is freely available online. It is cheaper than any other simulators.
- Any multifaceted network can be simulated and used for testing.
- The results can be obtained easily in the form of graph or in a network animator.
- It supports a wide variety of applications and protocols like TCP, UDP etc. which can be used for interaction between in the nodes.
- It can run on a variety of platforms like Windows, Linux, Unix etc.
- It is most widely used simulators.

1.3 Introduction to Proposed System

A WSN has multifarious applications in many fields. It is employed in many places. Ensuring the security in a WSN is of great concern. Because of the constraints in the network, it is susceptible to many attacks. The major attacks include denial of service, sinkhole, Sybil and hello flood attack. [4] These attacks decrease the performance and efficiency of the network. The attacks are studied in detail and are replicated in a simulator. The characteristics of the attack and the nature of the attack can be known. By simulating, the behavior of the network and the performance can be scrutinized. The network simulated is closer to real time network. By understanding the attacks, proper procedures can be taken in order to detect and prevent them. A simulator holds good for replicating the real time network. By understanding all the problems in the design phase itself, one can be able to construct a more efficient network.

2. PROPOSED SYSTEM DESIGN AND ARCHITECTURE

Figure 3.1 explains about the basic design of the sensor network and how all the nodes are connected in the network. The power generator supplies power to the power unit. The power unit supplies power to the sensing unit, processing unit and the transmission unit. Each sensor node is connected to a base station for communication by which it can send and receive data. It consists of a position finding system, mobilizer, sensing unit, processing unit and a transmission unit. BS indicates the base station.

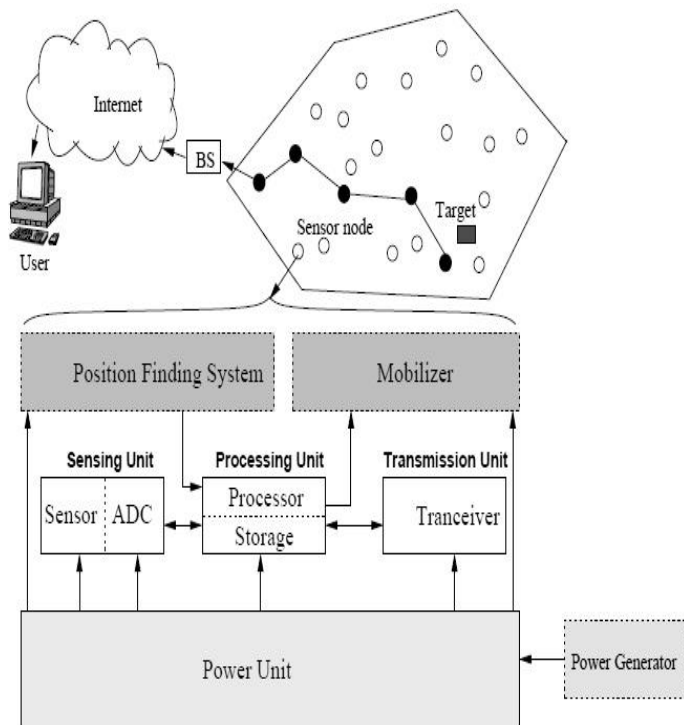


Fig 3.1: The components of a sensor node [1]

Figure 3.2 describes the details of the different layers in the wireless network and the communication process between the nodes and the wireless device. A WSN consists of an application layer, network layer, MAC and physical layer. The Sensor operates in the application layer. The packet is forwarded through a wireless channel from physical layer to application.

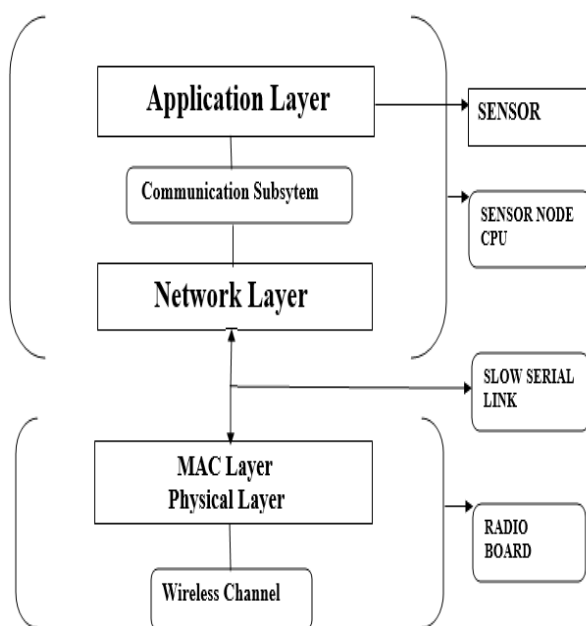


Fig 3.2: Layered architecture [20]

3. SYSTEM IMPLEMENTATION CONFIGURING NS

As said earlier WSN's are vulnerable to many attacks. Each attack may lead to a different problem. There are two types of attacks that are popular with the Wireless Sensor Networks. They are Physical attacks and logical attacks. [13] Physical attacks include capturing of the nodes and tampering the nodes which will lead to loss of data. On the other hand, Logical attacks include attacks like sinkhole attack, wormhole attack, hello flood attack, selective forwarding attack, Sybil attack, Denial of service attack. These attacks are more common in a Wireless Sensor Network. These attacks must be detected and must be avoided in order to increase the performance and security level in a WSN.

The Simulation of the attacks is being done by using NS2 Simulator. It is an open source free simulator available online. It stands good for simulation of TCP, UDP and many other routing protocols. It works on an object oriented language called Tool Command language (OTcl). With the help of OTcl language, different network topologies and the routing protocols can be explained. [14] The language is very easy to use and is platform independent. The code can be written for creation of the nodes, showing the data transfer and introducing the attacks and the simulation can be shown by running the simulator. The simulator consists of wide variety of applications, protocols like TCP, UDP and many other network parameters. The simulator consists of nodes and the data transfer between the nodes can be shown. The attacks can be introduced into the system by making some of the nodes malicious. In our system the simulation is shown on four attacks mainly Sybil attack, sinkhole attack, hello flood attack and denial of service attack.

3.1 Creating and Setting Connection between the Nodes in the Simulator

The first step is creation of the nodes in the network. Any number of nodes can be created in the network as per the user. The nodes can be made dynamic. The user can enter the source, destination and malicious node as he wishes when he runs the simulator. Figure 4.1 shows the creation of the nodes in the network. The movement of the nodes can be generated and the nodes can be partitioned into zones. After creating the nodes, a connection must be established between the nodes in the network. There are several protocols defined that can be used namely TCP and UDP. TCP is connection oriented protocol and it provides acknowledgement from the receiver. The UDP protocol can be used when there is a lot of traffic in the system which would be efficient. There is a TCP agent and a TCP sink. TCP agent is responsible for sending the packets in the network which can be called as a source node. TCP sink is the receiver node which receives the packets sent by the receiver.

Following shows how to create a node in the simulator. nn represents the number of nodes being initialized. The looping is done through the number and nodes and each node is created and assigned a random motion.

Code:

```
for{set i 0} {$i<$val(nn)} {incr i}
{set node_($i) [$ns node]
$node_($i) random-motion 1}
```

Following shows how to set up TCP connection between the nodes. gpstrace is a file that contains all the tcp connections in the network in data format. The file is opened in write mode and the tcp connection is set and the file is attached.

Code:

```
set gpstrace [open gpstrace.tr w]
set tcp [new Agent/TCP]
$tcp trace rtt_
$tcp trace cwnd_
$tcp attach $gpstrace
```

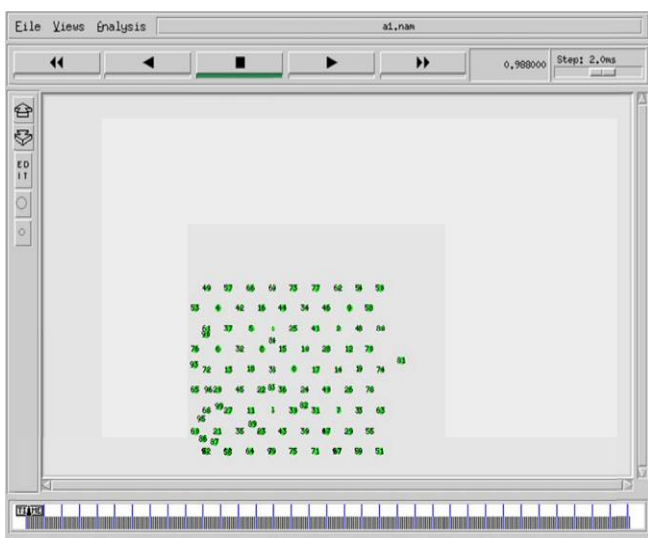


Fig 4.1: Creation of nodes in the simulator

The X dimension and the Y dimension of the topography in the system will be initialized. It represents the area in the simulator. The initial location of the nodes can be set at a particular coordinate in the simulator.

Following shows how to set the position of the nodes. The X coordinate is set to 20, whereas as the Y and Z are set to 0.

Code:

```
$node_(1) setX_20.000000000000
$node_(1) setY_0.000000000000
$node_(1) setZ_0.000000000000
```

3.2 Generation of Movements

Since it is a WSN, the nodes keep moving in the simulator. The generation of movements of the nodes in the simulator

can be done. The time at which the node should be moved to a particular destination can be set. Following shows the generation of movement of a node. The node 49 at time 0.2 sec moves to the particular destination.

Code:

```
$ns at 0.2 "$node_(49) setdest 100.78 980.56 3000"
```

3.3 Zone Partitioning

The nodes created can be partitioned into different zones. A color code can be assigned to each node. This is just for a clear view and identifying the nodes from each other. Following shows the partitioning of the nodes into zones and assigning a color code to them. The outline of the nodes 38, 74, 81 will be made pink at time 9.3 sec which is shown in figure 4.2.

Code:

```
$ns at 9.3 "$node_(38)
add-mark c3 pink circle"
$ns at 9.3 "$node_(74)
add-mark c3 pink circle"
$ns at 9.3 "$node_(81)
add-mark c3 pink circle"
```

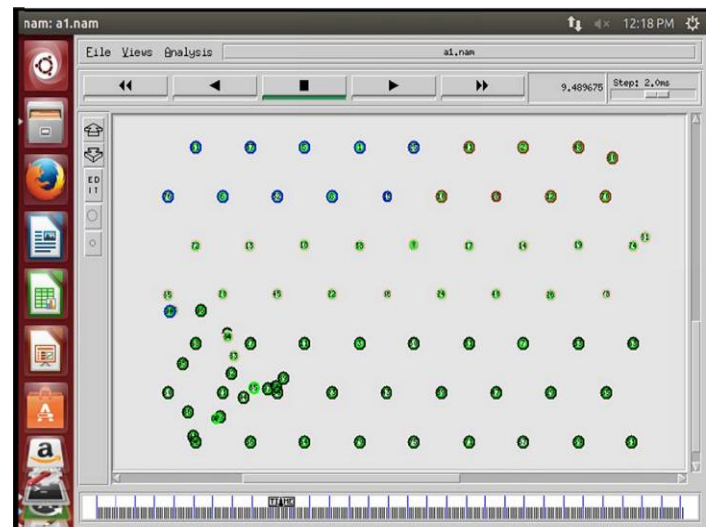


Fig 4.2: Partitioning into zones and assigning color codes.

3.4 Creating Application

The application indicates the type of transmission between the sender and the receiver. Here, constant bit rate (CBR) is used as an application in order to generate the traffic. The other applications that can be used are FTP, Telnet etc. The parameters for the application like time interval, maximum packets size etc are set. When the CBR application starts at a particular time, the packets will be sent from one node to the other node. Following shows how to set a CBR application to UDP. The parameters for the CBR application are initialized and the application is started.

Code:

```
set cbr_(1)[new Application/Traffic/CBR]
#Scbr_(1) set interval_2.0
```



```
#Scbr_(1) set random_1
#Scbr_(1) set maxpkts_100
#Scbr_(1) attach-agent $udp_(1)
$ns connect $udp_(1) $null_(1)
$ns at 76.0 "Scbr_(1) start"
$ns at 150.0 "Scbr_(1) stop"
```

3.5 Setting of Malicious Node

Any of the nodes created can be made as a malicious node to show different kinds of attacks. More than one malicious node can be created. The following shows how to set a particular node as malicious:

Code:

```
$ns at 50.0 "[node_ (30) set ragent_] malicious"
```

3.6 Making the Nodes Dynamic

The nodes created can be made dynamic which means the user can enter his/her own source, destination and malicious node to see the simulation happening between those nodes. It will help to make the simulation more interactive.

Following code shows how to make the nodes dynamic where argv0 entered from the terminal is set to source, argv1 is set to destination and argv2 is set to malicious node.

Code:

```
set src [lindex $argv 0]
set dest [lindex $argv 1]
set malicious [lindex $argv 2]
```

The user while running the file in the terminal along with the filename can give the nodes he wishes. Figure 4.3 shows the same where the user types `ns sinkhole.tcl 0 9 7` which means that the filename is `sinkhole.tcl`, the source is 0, the destination is 9 and the malicious node is 7.



Fig 4.3: User entering the nodes

3.7 Simulation of Cybil Attack

Sybil attack is one of the most harmful and dangerous attack in WSN. It is the attack in which a node acts as a malicious node and claims multiple identities. When there are many systems connected in a network, a single system which is insecure will act as a malicious system and claims multiple identities. This can lead to many problems like false communication and loss of data. This sort of an attack must be recognized and must be prevented so that the system can be made secure. Maintaining the identities of the system is necessary. There are many authorities that help in maintaining the identity by using certification software's [15]. Sybil attacks are the most common types of attacks. They tend to challenge the security and safety of the system. There are many ways to protect a system from Sybil attack. Trusted authority and proper identity can help prevent a network from such type of an attack.

The simulation of the Sybil attack is done by using the NS2 Simulator. It can be done by modifying `aodv.cc` file in `ns2.35` which can be shown by dropping the packets in the simulator. Figure 4.4 shows the simulation of the Sybil attack. The attack can be seen by dropping of the packets of the intermediate node. This attack is one of the well-known attack in WSN. The attacker nodes may obtain the legitimates IP Address or Mac Address in order to Steal and make its own. Then the attacker node can do plenty of things with new stolen identity. Node 43 acts as source whereas node 44 is the destination node. The source node start sending packets to the destination node through the shortest path that is decided by the routing protocol. The intermediate node 15 acts as a malicious node and at time 30 sec, it starts dropping the packets coming from the node.

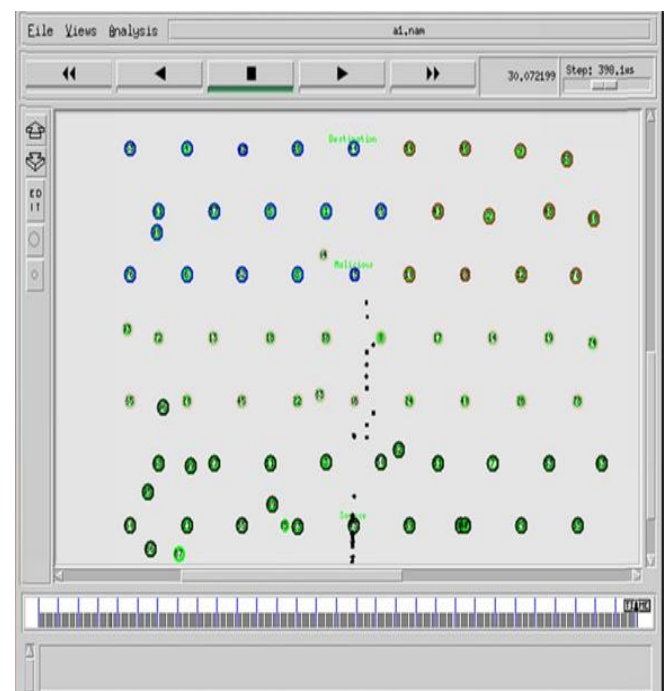


Fig 4.4: Simulation of Sybil attack

3.8 Simulation of Denial of Service Attack

WSN are concerned with numerous security issues. The constraint on the resources makes the WSN more vulnerable to Denial of Service attack because it focusses on the energy protocols. [16] DoS attack prevents the system or the user to be legitimate. It can be done by overloading the destination system with huge number of requests. Due to this attack, efficiency and the performance of the Wireless Sensor Network would be reduced. This particular type of attack in unfriendly situations can be even more harmful. There are many types of DOS attacks. Among the different types of DOS Attacks, SYN flood is the most common kind of an attack. It uses TCP three way handshake mechanism for communication between the nodes.

Denial of service Attack involves saturating the performance of the target node with lots of unwanted communication requests which will create fake traffic. [16] These kinds of attacks overload the server. Here, DOS attack is implemented by using UDP protocol and CBR application. Once its buffer size is full, the target node can be seen dropping the packets coming from the malicious node as well as the source. Figure 4.5 shows the simulation of the denial of service attack. Node 41 is the source and node 50 is the destination. The packets from the source node are sent to the destination node via the target node 58. After sometime node 48 acts as a malicious node and starts sending huge number of packets to the target node. Since the target node buffer size is limited, it cannot handle all the packets and at time 23 sec will drop the packets coming from the malicious node 48 as well as the source node 41. This will lead to the loss of data and will degrade the service of the network.

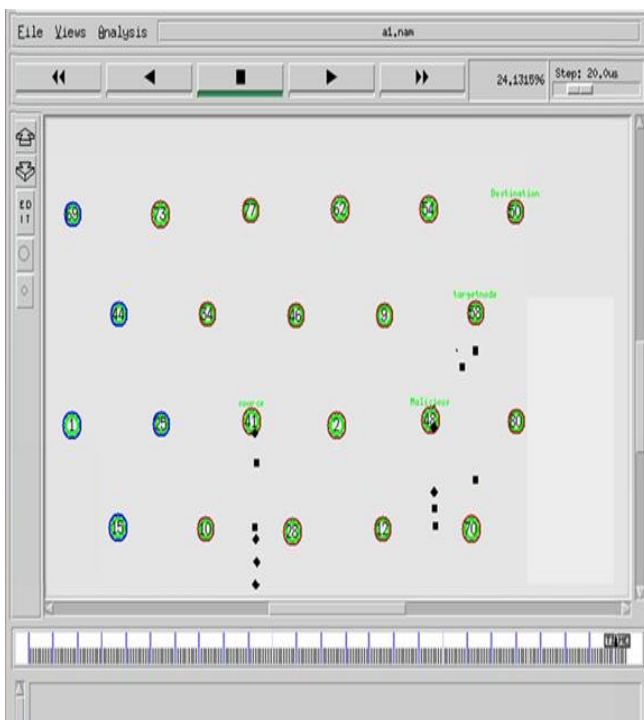


Fig 4.5: Simulation of DOS attack

3.9 Simulation of Sinkhole Attack

A WSN consists of many nodes connected to each other. These nodes would send the collected data to the base station for processing. A sinkhole attack is more seen in cases where there is many to one communication. It is a serious threat to the sensor networks and proper measures should be taken in order to detect and prevent it [18]. Here, a malicious node acts as destination node and looks attractive to the surrounding nodes. Sinkhole attack is the selective forwarding attack. The malicious node will be closer to the destination node in order to attract all maximum possible traffic of the network. It is one of the complex attack and detection of the Sinkhole attack is very difficult. In the simulator, the malicious node is placed beside the destination node and it attracts all the packets instead of forwarding to the legitimate node. There are some routing protocols that could withstand the sinkhole attack at a certain level but many of the current ones are affected by the sinkhole attack. Figure 4.6 shows the simulation of sinkhole attack. In the initial stage, the malicious node gets all the information about the neighbor nodes. The node that is closer to the destination acts as a malicious node. The user can enter the source, destination and the malicious node in the command prompt when he runs the simulation. Figure 4.12 shows the user entering 0 as source, 9 as destination and 7 as malicious node.

```

sheranusha@ubuntu: ~/Documents
sheranusha@ubuntu:~$ export TCL_LIBRARY=/home/sheranusha/Downloads/ns-allinone-2.35/tcl8.5.10/library:/home/sheranusha/Downloads/ns-allinone-2.35/tk8.5.10/library
sheranusha@ubuntu:~$ cd Documents
sheranusha@ubuntu:~/Documents$ ls
DosAttack.tcl sinkhole.tcl Sybil.tcl
sheranusha@ubuntu:~/Documents$ ns sinkhole.tcl
num_nodes ls set 10
INITIALIZE THE LIST xListHead
can't read "node_()": no such element in array
-----while executing-----
"$ns_ at 1.0 "$node_($src) label Source"
(file "sinkhole.tcl" line 122)
sheranusha@ubuntu:~/Documents$ ns sinkhole.tcl 0 9 7
num_nodes ls set 10
INITIALIZE THE LIST xListHead
Starting Simulation...
channel.cc:sendUp - Calc highestAntennaZ_ and distCST_
highestAntennaZ_ = 1.5, distCST_ = 550.0
SORTING LISTS ...DONE!
NS EXITING...
sheranusha@ubuntu:~/Documents$ nan a1.nam

```

Fig 4.6: User making the nodes dynamic

As entered by the user, node 0 becomes the source node, node 9 becomes the destination node and node 7 is made a malicious node. The malicious node acts as a legitimate node and attracts all the maximum possible traffic in the network. All the packets being sent from the source are sent to the malicious node instead of the destination node which can be seen in the figure 4.7.

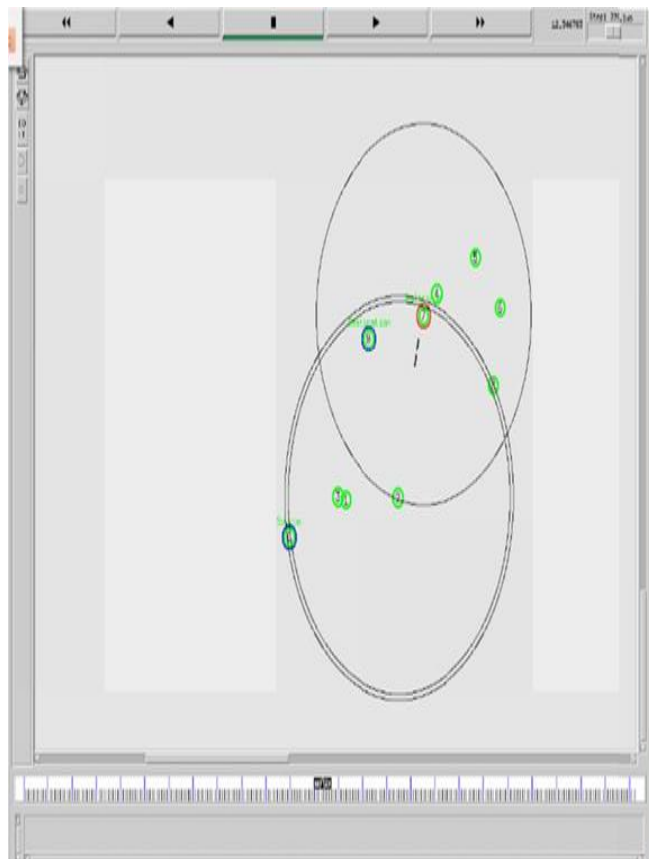


Fig 4.7: Simulation of sinkhole attack

3.10 Simulation of Hello Flood Attack

Hello Flood attack is also one of the most common attacks in a WSN. In this type of attack a malicious node keeps sending hello requests to the legitimate node which will alter the security of the system. [17] The node which receives such a message assumes that it has been sent by the sender which is not the case always. It can occur when there is huge amount of traffic in the system. Several cryptographic techniques and methods have been implemented in order to prevent this attack but each one had its own drawback.

This attack can be simulated by modifying the Aodv.h and Aodv.cc file in ns2 simulator in order to create hello flood attack where we can see the target node being flooded by the packets. These files are the inbuilt files that come along with the ns2 package when one downloads. They contain all the code about the routing, providing a path for routing and information on the packet forwarding. Figure 4.8 shows the simulation of hello flood attack. A node is made as a target node and it is flooded with lots of hello messages which will create a lot of black circles in the simulator. The user can enter the source and destination as he wishes which is

shown in figure 4.9. Here, the node 0 is made as the source and the node 9 is made destination. Every node is seen sending hello messages to every other nodes in the network.

```

sheranusha@ubuntu: ~/Documents
sheranusha@ubuntu:~$ export LD_LIBRARY_PATH=/home/sheranusha/Desktop/ns-allinone-2.35/otcl-1.14:/home/sheranusha/Desktop/ns-allinone-2.35/lib
sheranusha@ubuntu:~$ export TCL_LIBRARY=/home/sheranusha/Desktop/ns-allinone-2.35/tcl8.5.10/library:/home/sheranusha/Desktop/ns-allinone-2.35/tk8.5.10/library
sheranusha@ubuntu:~$ export PATH=SPATH:/home/sheranusha/Desktop/ns-allinone-2.35/bin:/home/sheranusha/Desktop/ns-allinone-2.35/tcl8.5.10/unix:/home/sheranusha/Desktop/ns-allinone-2.35/tk8.5.10/unix
sheranusha@ubuntu:~$ cd Documents
sheranusha@ubuntu:~/Documents$ ls
a1.nam      DosAttack.tcl- Helloflood.tcl  Sybil.tcl      tracebr.tr
DosAttack.tcl  gprstrace.tr  sinkhole.tcl   Sybil.tcl-    tracesh.tr
sheranusha@ubuntu:~/Documents$ ns Helloflood.tcl 0 9
num_nodes is set 100
warning: Please use -channel as shown in tcl/ex/wireless-mif.tcl
INITIALIZE THE LIST xListHead
channel.cc:sendUp - Calc highestAntennaZ_ and distCST_
highestAntennaZ_ = 1.5, distCST_ = 550.0
SORTING LISTS ...DONE!
end simulation
sheranusha@ubuntu:~/Documents$
    
```

Fig 4.8: User making the nodes dynamic

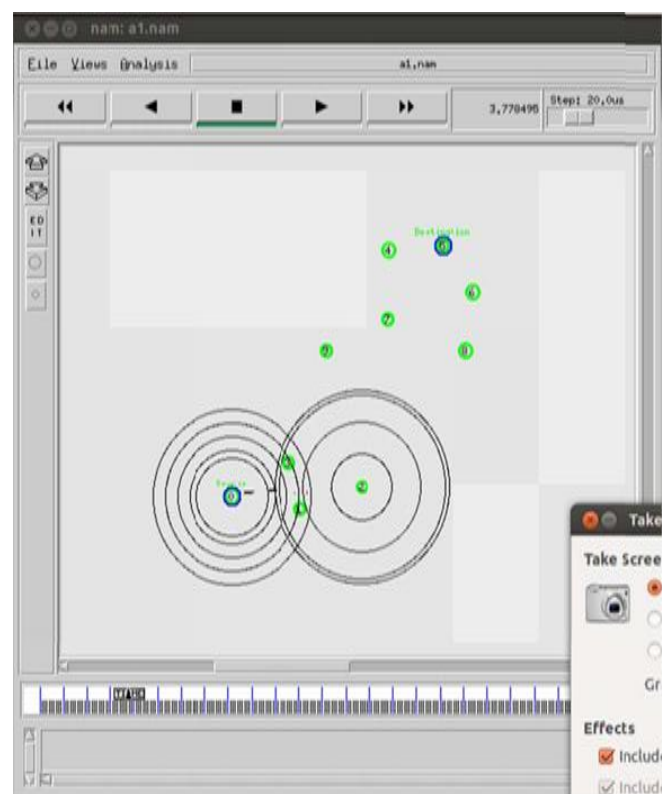


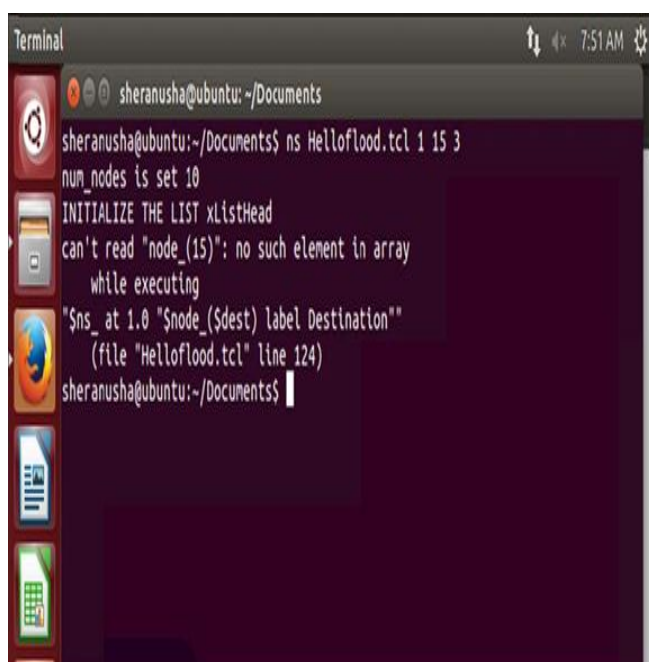
Fig 4.9: Simulation of hello flood attack

4. TESTING AND EVALUATION

Using the network simulator NS2, the attacks can be simulated. It creates a replica of a real time network [12]. NS Simulators are mainly used for network research and learning. It helps me to create security nodes and establish the communication between them. This NS2 tool allows to develop a model design for WSN connection between nodes in the network. Based on the network attacks like Dos, wormhole attack, hello flood attack, sinkhole attacks, Sybil attack, selective forwarding attacks, the network security can be tested [9]. NS2 tool gives scope in testing, so the level of network security can easily be tested. The attacks can be created in the network and the security level of the wireless sensor network can be tested. Here, for the given input, what output it is generating are my test cases.

Test Case 1: When the node entered by the user is greater than the number of nodes:

From the terminal, the user can enter his own source, destination and malicious node. A fixed number of nodes will be created. So, if the entered node is greater than the number of nodes created, the simulation will not start because it cannot read the node. In the figure 5.1, the number of nodes created are 10. The user has entered 15 as destination node. Since it cannot find the node for communication, it throws an error.

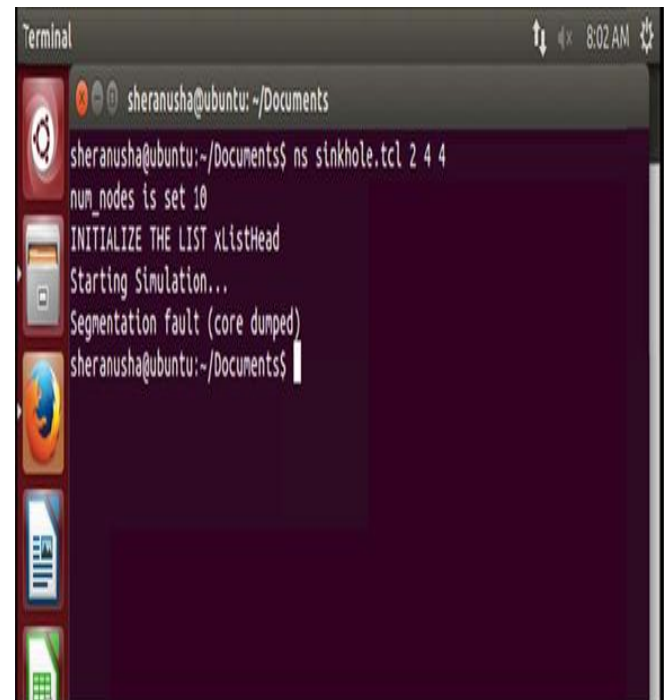


```
Terminal
sheranusha@ubuntu:~/Documents
sheranusha@ubuntu:~/Documents$ ns Helloflood.tcl 1 15 3
num_nodes is set 10
INITIALIZE THE LIST xListHead
can't read "node (15)": no such element in array
while executing
"$ns at 1.0 "$node_($dest) label Destination"
(file "Helloflood.tcl" line 124)
sheranusha@ubuntu:~/Documents$
```

Fig 5.1: Incorrectly entered node

Test Case 2: Overlapping of nodes

It can be tested with sinkhole attack. In sinkhole attack, a node that is closer to destination node will act as malicious node. If the user gives the same node for destination as well as malicious node, the attack will not happen. It will overlap the transmission and will create a segmentation fault. Figure 5.2 shows the same.



```
Terminal
sheranusha@ubuntu:~/Documents
sheranusha@ubuntu:~/Documents$ ns sinkhole.tcl 2 4 4
num_nodes is set 10
INITIALIZE THE LIST xListHead
Starting Simulation...
Segmentation fault (core dumped)
sheranusha@ubuntu:~/Documents$
```

Fig 5.2: Overlapping of nodes

Test Case 3: Choosing the routing path

In Sybil attack, lot of transmissions between the nodes are created. Each transmission will have its own routing path. The routing path will be decided by the AODV file that comes with the installation of ns2 package. For the Sybil attack to happen, the malicious node should be one of the intermediate nodes between the source and destination. If the malicious node created resides in the path between the source and destination, then the attack occurs. It can be shown in figure 5.3.

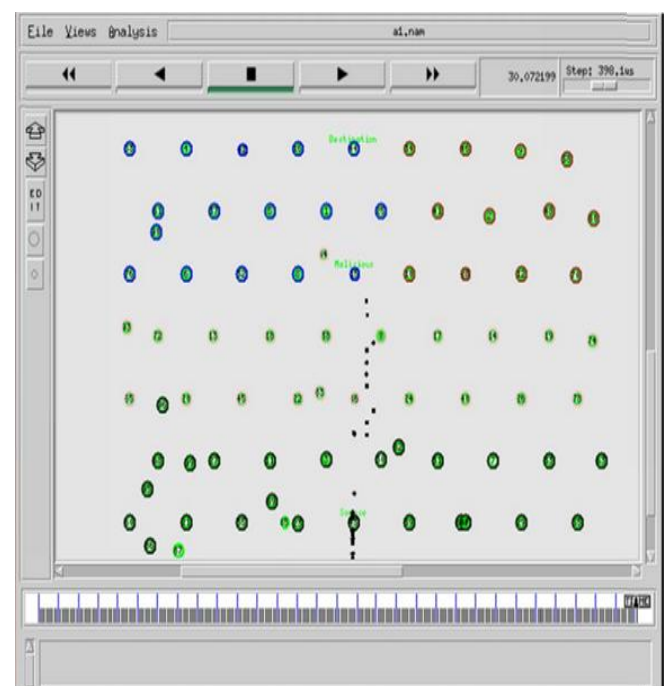


Fig 5.3: Simulation of Sybil attack

5. RESULTS

As a part of the proposed system, a graph has been generated to monitor the network performance. It calculates the throughput of the network. Throughput indicates overall number of bytes received in the network. The Throughput in the network can be affected by various number of factors. It plays a vital role in analyzing the network performance. The trace file generated is passed as an input in order to generate a graph. Using the graph, one can easily understand the simulation results of the network. The X Axis represents the time and Y Axis represents the throughput rate. Initially, when the transmission starts there is huge amount of traffic in the system, so the throughput will be high. Later it drops at one point when an attack occurs. Figure 5.4 shows the graph.

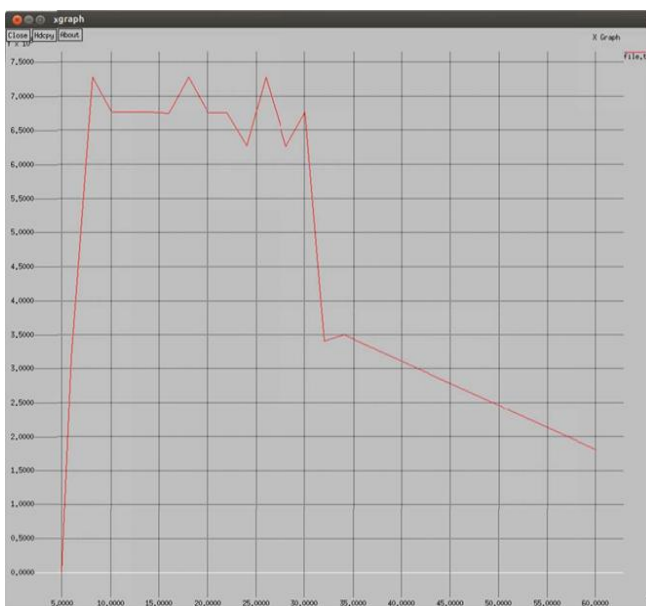


Fig 5.4: Graph for throughput

6. CONCLUSIONS AND FUTURE WORK

WSN's are of huge demand. The request for wireless sensor networks are increasing rapidly, because the growth of using WSN has increased. There are some limitations in a wireless sensor network like they have limited storage capacity, limited capability of processing and limited energy to transmit data [1]. These drawbacks can make WSN different from other networks. There are some little concerns that occur in a WSN. Based on the above mentioned difficulties in the data integrity, security, there are many solutions that are available to overcome these dangers. The attacks that are popular in a WSN like hello flood attack, sinkhole attack, Sybil attack and denial of service attack have been simulated in a simulator. On simulation, the performance and the efficiency of the network can be analyzed. The behavior and the energy parameters can be examined. A mechanism for ensuring secure data transfer and preventing the attacks in a WSN must be proposed. The parameters which determine the network performance can be calculated from the simulation. Because of the numerous attacks happening in the WSN, there is less amount of security.

REFERENCES

- [1]. Y. Wang, G. Attebury, et al. "A survey of security issues in wireless sensor networks." Computer Science and Engineering. Vol.8, no. 2. 2006.
- [2]. E. Shi and A. Perrig, "Designing Secure Sensor Networks," Wireless Commun. Mag., vol. 11, no. 6, pp. 38–43, Dec. 2004.
- [3]. N. Gura, A. Patel, et al. "Comparing elliptic curve cryptography and RSA on 8-bit CPUs." Cryptographic Hardware and Embedded Systems-CHES 2004, pp 925-943, 2004.
- [4]. M. Razzaque., S.Ahmad Salehi. Security and Privacy in Vehicular Ad- Hoc Networks: Survey and the Road Ahead. Wireless Networks and Security, Springer: 107-132, 2013.
- [5]. A. Perrig et al., "SPINS: Security Protocols for Sensor Networks," Wireless Networks, vol. 8, no. 5, pp. 521–34, Sept. 2002.
- [6]. H. Du, X. Hu, et al. "Energy efficient routing and scheduling for realtime data aggregation in WSNs." Computer communications. Vol.29, no. 17. 3527-3535, 2006.
- [7]. X. Hung, et al. "An Energy-Efficient Secure Routing and Key Management Scheme for Mobile Sinks in Wireless Sensor Networks Using Deployment Knowledge," Sensors, Vol 8. 2008, 7753-7782
- [8]. L. Jiali, Valois, F.; Dohler, M.; Min-You Wu; "Optimized Data Aggregation in WSNs Using Adaptive ARMA," Sensor Technologies and Applications (SENSORCOMM), 2010 Fourth International Conference on pp.115-120, 18-25 July 2010.
- [9]. S. Zhu et al., "An Interleaved Hop-by-Hop Authentication Scheme for Filtering of Injected False Data in Sensor Networks," Proc. IEEE Symp. Security and Privacy, Oakland, CA, pp. 259–71, May 2004.
- [10]. J. Ben-Othman, and B. Yahya. "Energy efficient and QoS based routing protocol for wireless sensor networks." Journal of Parallel and Distributed Computing 70(8), 849-857 2010.
- [11]. D.W. Carman, P.S. Krus, and B.J. Matt, "Constraints and approaches for distributed sensor network security", Technical Report 00-010, NAI Labs, Network Associates Inc., Glenwood, MD, 2000.
- [12]. Teerawat Issariyakul and Ekram Hossain. Introduction%20to%20Network%20Simulator%20NS2%20(1).pdf
- [13]. R. E. Shannon, "Introduction to the art and science of simulation," in Proc. of the 30th conference on winter simulation (WSC'98), 1989 [14] <http://ns2tutor.weebly.com/ns2-in-windows.html>
- [15]. <http://www.cs.berkeley.edu/~dawnsong/papers/sybil.pdf>
- [16]. <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4431860>
- [17]. <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.403.6592&rep=rep1&type=pdf>
- [18]. <http://ijaiem.org/Volume2Issue2/IJAIEM-2013-02-06-005.pdf>
- [19]. <http://sourceforge.net/projects/nsnam/files/>
- [20]. <http://www.ipcsit.com/vol35/003-CNCS2012-N010.pdf>

BIOGRAPHIES



Tejaswi Singh, a student of computer science engineering in his final year with a knack for network security.



Aatish Gandotra, a student of Electronics and Communication Engineering in his final year with deep interest in telecommunication and networking.