A NOVEL TECHNIQUE FOR SECURE ROUTING IN WIRELESS SENSOR NETWORKS

A Major Project Thesis
Submitted in the partial fulfillment of the requirements for
the award of the degree of

Bachelor of Technology
in
Department of Computer Science and Engineering

By

Deevela Adithya (Roll no:150030227)
Kakani Srihitha (Roll no:150030368)
Vasireddy Sowmya (Roll no:150030973)

under the supervision of
Mrs. K. Swetha
ASSISTANT PROFESSOR,CSE DEPT.

Department of Computer Science and Engineering

K L E F, Green Fields,

Vaddeswaram- 522502, Guntur(Dist.), Andhra Pradesh, India.

April,2019.

KONERU LAKSHMAIAH EDUCATION FOUNDATION

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Declaration

The Major Project Report entitled "**A NOVEL TECHNIQUE FOR SECURE ROUTING IN WIRELESS SENSOR NETWORKS**" is being submitted by D. Adithya (150030227), K. Srihitha (150030368), V. Sowmya (150030973) submitted in partial fulfillment for the award of Bachelor of Technology in **Computer Science and Engineering** during the academic year 2018-2019.

Submitted by

Deevela Adithya	(Roll no:150030227)
Kakani Srihitha	(Roll no:150030368)
Vasireddy Sowmya	(Roll no:150030973)

KONERU LAKSHMAIAH EDUCATION FOUNDATION

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Certificate

This is to certify that the Minor Project Report entitled "**A NOVEL TECHNIQUE FOR SECURE ROUTING IN WIRELESS SENSOR NETWORKS**" is being submitted by D. Adithya (150030227), K. Srihitha (150030368), V. Sowmya (150030973) submitted in partial fulfillment for the award of Bachelor of Technology in **Computer Science and Engineering** during the academic year 2018-2019.

Signature of the Supervisor	**Signature of the HOD**
Mrs. K. Swetha	**Mr. V. Hari Kiran**
ASSISTANT PROFESSOR	ASSOCIATE PROFESSOR

Signature of the External Examiner

ACKNOWLEDGEMENTS

Our sincere thanks to **K. Swetha** Madam, Major Project Guide for his outstanding support and guidance throughout the project for the successful completion of the work. We express our gratitude to **Mr. Hari Kiran Vege,** Head of the Department for Computer Science Engineering, for providing us with adequate facilities, ways and means by which we can complete this term paperwork.

We would like to place on record the deep sense of gratitude to the **Honorable Vice Chancellor**, KLEF for providing the necessary facilities to carry the concluded term paperwork.

Last but not the least, we thank all Teaching and Non-Teaching Staff of our department and especially our teammates and friends, for their support in the completion of our project work.

ABSTRACT

Wireless Technology is at its peak when we talk about research and innovation. This field has become a hub of invention of new theories and structures. Mobile Ad-hoc Network is a special point of focus for the researchers. MANET is a collection of wireless mobile nodes which forms a dynamic temporary network without using any existing infrastructure. In recent time the market of mobile devices, laptops, handheld and portable devices is at its zenith so communication between such devices is an important issue. Routing is an essential part in the success of communication among these mobile structures. Routing protocols play an important role for finding an efficient and reliable route from source to destination. In the literature, there are numerous MANET routing protocols aiming to find the most suitable path from source to destination. In this paper, a comprehensive simulation-based performance study and analysis is performed on various types of routing protocols over MANET. Ad Hoc On-Demand Distance Vector (AODV), Dynamic Source Routing (DSR), Temporally-Ordered Routing Algorithm (TORA), Optimized Link State Routing (OLSR) and Geographic Routing Protocol (GRP) has been considered for investigation in this paper based on OPNET simulation. Moreover, the performance of these routing protocols will be measured based on throughput, delay, load and data dropped metrics.

INDEX

CHAPTER-1

INTRODUCTION

Network structure is changing rapidly in recent years. The only network available four decades ago was wired network. The emergence of wireless networks has gone a long way in solving the growing service demands. The focus of research and development has almost shifted from wired networks to wireless networks. The limitations of wireless network techniques such as high error rate, power restrictions, bandwidth constraints, etc has not deterred the growth of wireless networks. Mobile Ad-hoc network (MANET) is the most demanding field in the area of wireless networks. MANET has mobile devices or users which are generally known as nodes each one of which is equipped with radio transmitter and receiver. MANET is a temporary network of wireless mobile nodes which has no fixed infrastructure. There are no dedicated routers, servers, access points, base stations and cables. The mobile nodes which are within each other’s transmission range can communicate with each other directly otherwise, the nodes in between them forward the packets for them from source to destination. Every node act as a router to forward the packets to other nodes whenever required. One of the main areas of research has been routing technology which will route packets from source to destination. In this paper, we have evaluated performance of AODV, DSR, TORA, OLSR and GRP routing protocols based on e-mail and video conferencing traffic generating applications by increasing the number of nodes.

Chapter 2

PROBLEM STATEMENT

DATA SECURITY IN WIRELESS SENSOR NETWORKS

Security of the information can be estimated as far as Availability, Confidentiality and Integrity. Accessibility guarantees that the information is accessible for access to the clients whenever and from anyplace through the web. Secrecy guarantees the clients that their information is protected from unapproved get to. Though, the objective of information trustworthiness is to guard the information from any unapproved changes.

In wireless sensor networks, security of data plays an important role. But nowadays so many attacks are taking place on wsn’s. So inorder to avoid those attacks we will see some countermeasures in this paper.

PROBLEM FORMULATION OBJECTIVES

- Ensuring Confidentiality of the data
- Ensuring security of the data stored
- Security of the data in transit
- Effective retrieval of encrypted data
- Ensuring that there is no plain text leak out.

CHAPTER-3

LITERATURE SURVEY

In this paper, we are finding a solution for attacks that are faced by wireless sensor networks. First to understand this paper, we have also read some IEEE journals related to this paper. Some of them are:[1] Wireless Sensor network attacks and security mechanisms, in this paper they have discussed attacks faced by wsn's and also some measures for that. But the disadvantage of this paper is they could not find the solutions for all the attacks.[2] key management schemes in wireless sensor networks, in this paper we will explain different key management schemes, critique them theoretically, and propose an idea as a way out for the expected problems in one of these schemes.. They have used key management techniques like RSA and ECC.[3] Dynamic key management in wireless sensor networks, in this paper they have described an emerging class of schemes, dynamic key management schemes, assumes long-lived networks with more frequent addition of new nodes, thus requiring network rekeying for sustained security and survivability.

CHAPTER-4

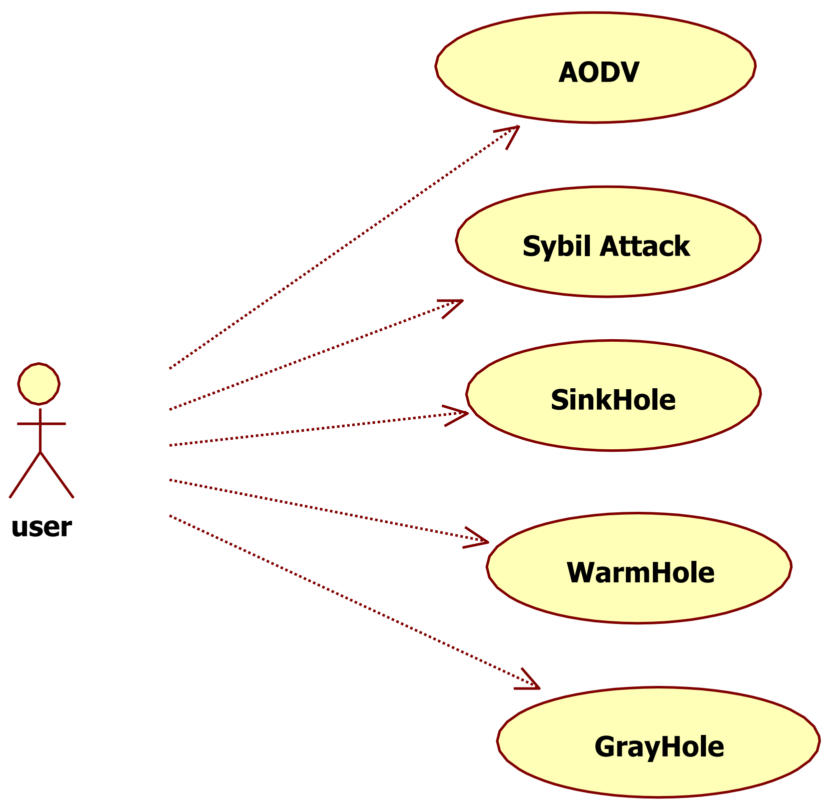
THEORETICAL ANALYSIS

MANET ROUTING PROTOCOLS

There are several protocols proposed for wireless mobile ad-hoc networks. The first three protocols are selected from Reactive category namely AODV, DSR and TORA and the fourth is selected from proactive category namely OLSR whereas the last protocol is selected from Geographic Position Information based routing namely GRP.

Ad-hoc On Demand Distance Vector (AODV)

AODV is reactive routing protocol which does not discover or maintain a route until or unless requested by nodes. AODV uses destination sequence number to ensure the loop freedom and freshness of route. AODV is capable of both unicast and multicast routing. The operation of protocols is divided into two functions: route discovery and route maintenance. When a node requests to communicate with another node it starts route discovery mechanism. The source node sends a route request message RREQ to its neighbors and if all those neighbor nodes do have any information about the destination node then they will further send the message to its neighbors and so on until the destination node is node is found. The node which has information of the destination node sends a route reply message RREP to the initiator of the RREQ message. The path is recorded in the intermediate nodes in the routing table and this path identifies the route. When the initiator receives the route reply message the route is ready and the initiator can start sending the packets. The route error RRER is reported when the link with the next hop breaks.



Dynamic Source Routing (DSR)

DSR is also a reactive routing protocol which uses the concept of source routing. In source routing the sender knows complete hop-by-hop route to the destination. All the routes are stored in the route cache. When a node attempts to send a data packet to a destination for which it does not know the route . In DSR each node maintains a route cache with route entries which are continuously updated as and when route learns new routes. The biggest advantage of DRS is that no periodic routing packets are required. DSR has also the capability to handle unidirectional links . Unlike other protocols DSR requires no periodic packets of any kind at any layer within the network. The sender of the packets selects and controls the route used for its own packets, which also supports features such as load balancing. All routes used are guaranteed to be free of loops as the sender can avoid duplicate hops in the selected routes.

Temporally Ordered Routing Algorithm (TORA)

TORA is an adaptive on demand routing protocol for multi hop networks. TORA is source initiated specially proposed routing protocol for highly dynamic mobile, multi -hop wireless networks [8]. TORA is based on link reversal algorithms. TORA establish the routes quickly and minimize the communication overhead by localizing algorithm reaction to topological changes when possible [10]. Instead of using the concept of shortest path for computing routes which take huge amount of bandwidth TORA algorithm maintains the “direction of the next destination” to forward the packets. Thus the source node maintains one or two “downstream paths” to the destination node through multiple intermediate neighboring nodes. The three steps involved in TORA are: a) route creation, b) route maintenance, and c) route erasure. TORA uses the concept of “directed acyclic graphs” to establish downstream paths to destination and such DAG is known as “Destination Oriented DAG”.

Optimized Link State Routing (OLSR)

OLSR is a proactive routing protocol, in which all routes have route table for maintaining information to every node in the network. The routes are immediately available whenever needed due to the route tables. OLSR is an optimized version of link state protocol. OLSR uses the concept of Multipoint Relays (MPR) to reduce the possible overhead in the network. OLSR uses two types of control messages: Hello and Topology Control (TC). Hello message are used to find the link state and neighboring nodes. TC message is used to for broadcasting information for own advertised neighbors which includes at least the MPR selector list.

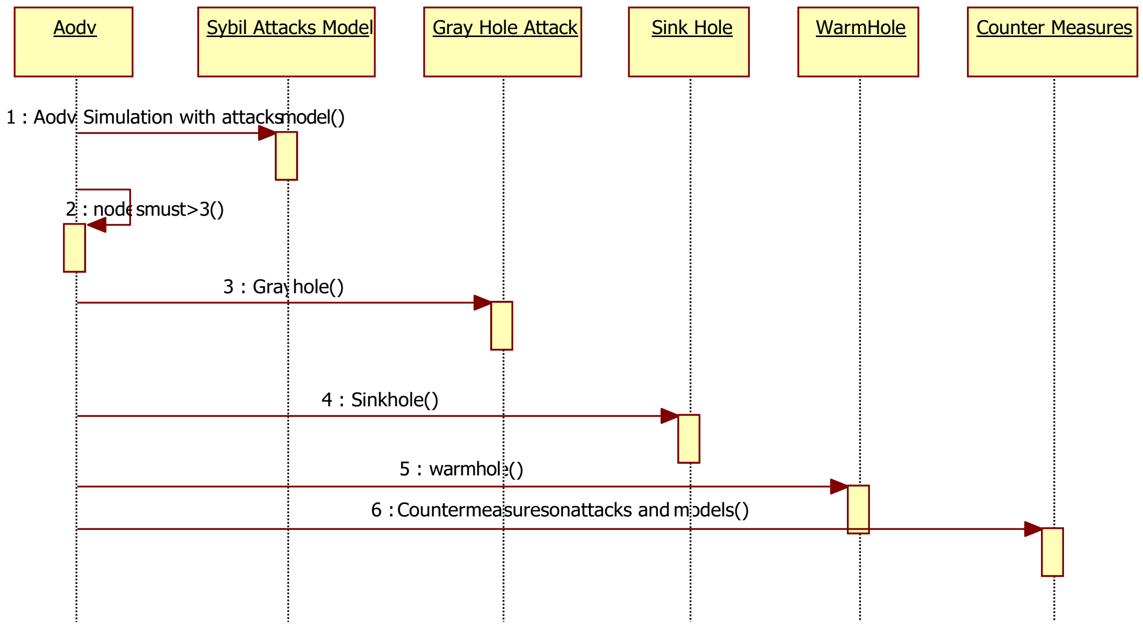
Geographic Routing Protocol (GRP)

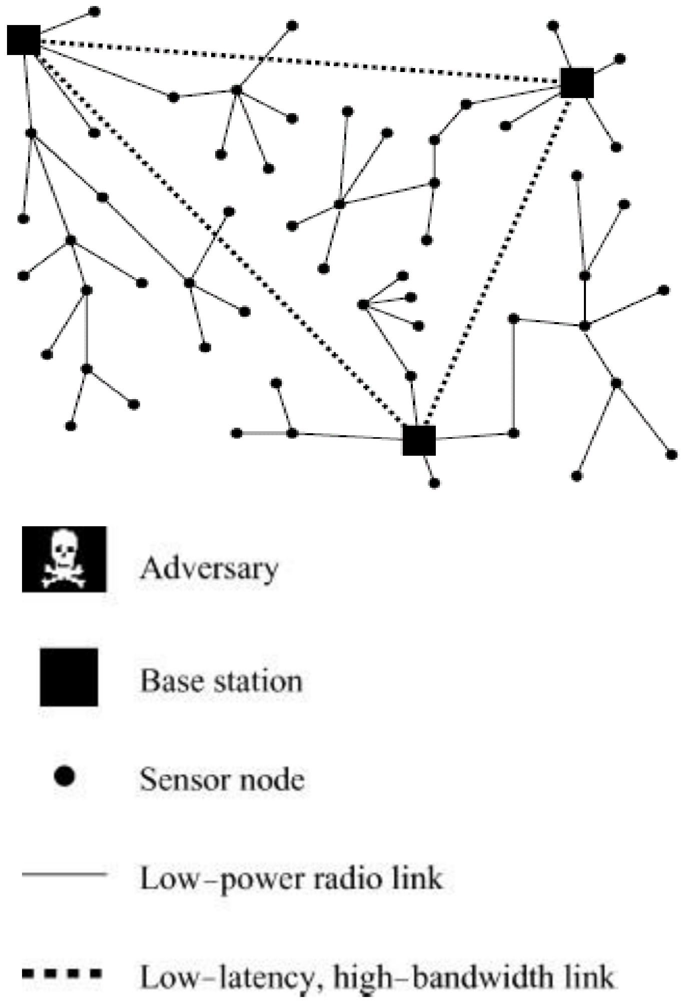
GRP is a position based protocol classified as proactive routing protocol. In GRP the Global Positioning System (GPS) is used to mark the location of node and thee quadrants optimize flooding. When a node moves and crosses neighborhood then the flooding position is updated. The neighbors and their positions are identified by the exchange of “Hello” protocol. The concept of route locking ensures that a node can return its packet to the last node when it cannot keep on sending the packet to the next node. The network is divided into quadrants to reduce route flooding. The entire world is divided into quadrants from Lat, Long

(-90, -180) to Lat, Long (+90, + 180) [12]. Apart from actual geographic coordinates received by the GPS the other approach.

Threat Models:-

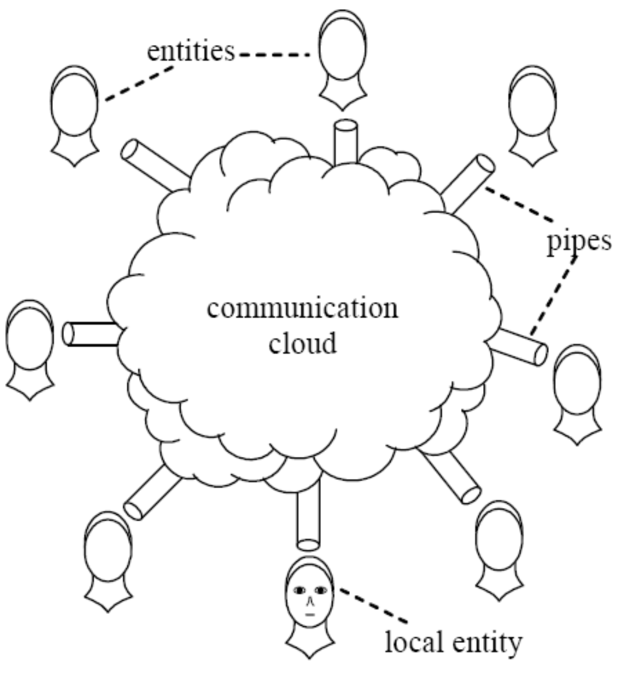
- ✓ Threat models and security goals for routing in WSNs
- ✓ Two new attacks
  - ❑ Sinkhole attacks
  - ❑ HELLO floods
- ✓ How to adapt attacks against ad-hoc wireless networks into powerful attacks againstWSNs
- ✓ Practical attacks against routing protocols and topology maintenance algorithms of WSNs
- ✓ Countermeasures and design considerations for secure routing protocols in WSNs
- ✓ WSNs consist of hundreds or thousands of low-power, low-cost nodes having a CPU, power source, radio, and other sensing elements
- ✓ Have one or more points of centralized control called base stations or sinks
- ✓ Sensor readings from multiple nodes processed at aggregation points
- ✓ Power is the scarcest resource





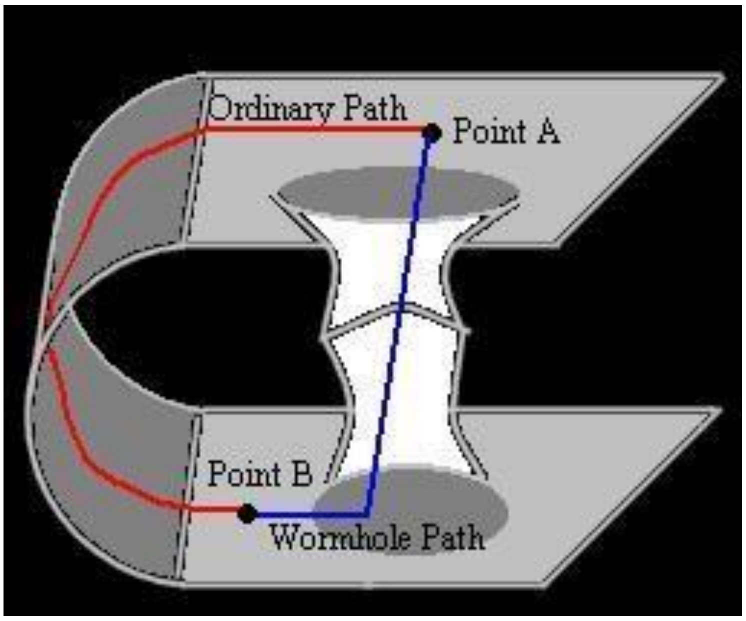
Sybil Attack:-

One can have, some claim, as many electronic persons as one has time and energy to create.”



Wormhole:-

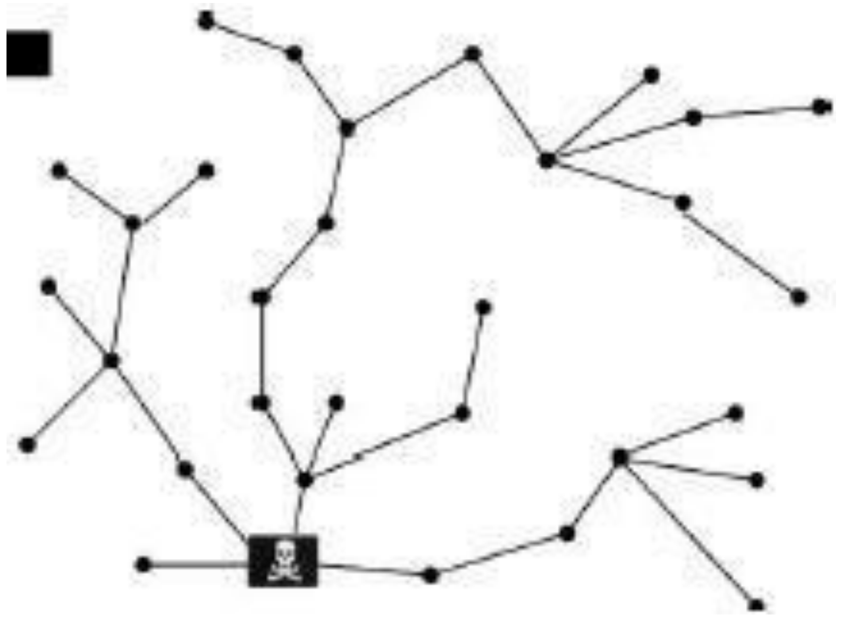
An adversary tunnels packets received in one part of the network over a low-latency link and replays them in a different part of the network



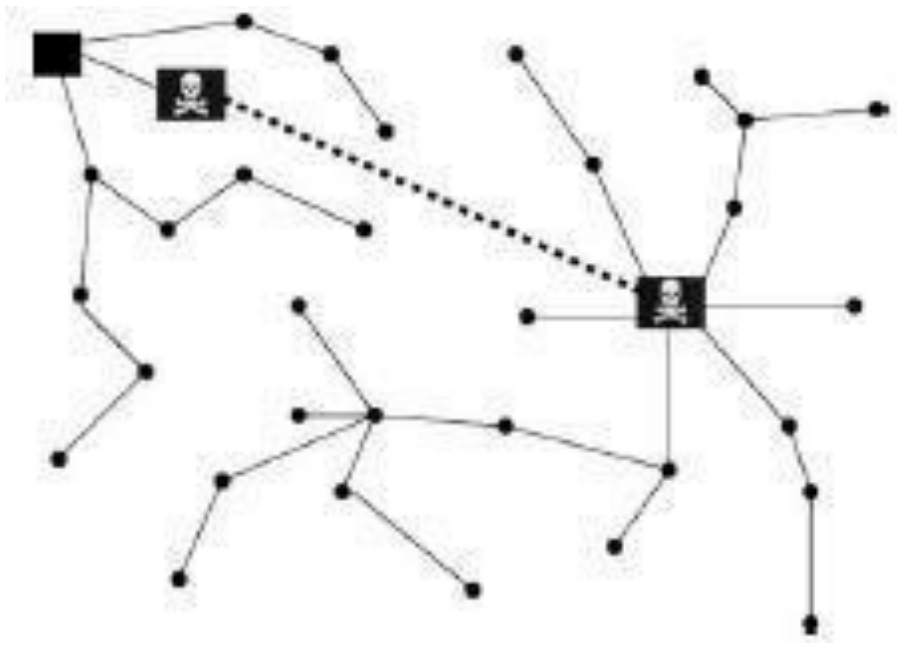
Flood attack:-

- ❑ Many protocols require that nodes broadcast HELLO packets to announce themselves to their neighbors
- ❑ Laptop-class attacker can convince all nodes that it is their neighbor by transmitting at high power

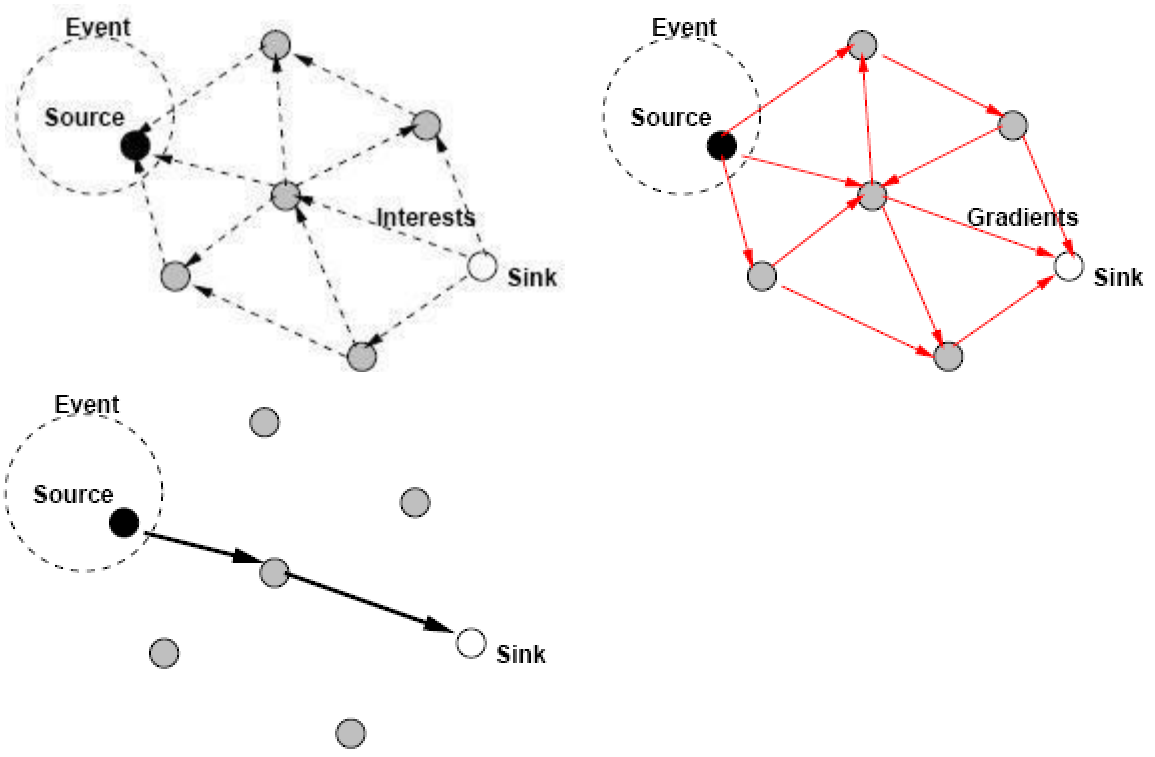
- ✓ Acknowledgement spoofing
- ✓ Tiny OS becoming
    - ❑ Description
    - ❑ Attacks
- ✓ Can authenticated routing updates solve the problem?



- Combined wormhole/sinkhole attack



Directed diffusion:



- Attacks – Suppression, Cloning, Path influence, Selective forwarding and data tampering.

Protocol	**Relevant attacks**
TinyOS beaconing	Bogus routing information, selective forwarding, sink-holes, Sybil, wormholes, HELLO floods
Directed diffusion and its multipath variant	Bogus routing information, selective forwarding, sink-holes, Sybil, wormholes, HELLO floods
Geographic routing (GPSR, GEAR)	Bogus routing information, selective forwarding, Sybil
Minimum cost forwarding	Bogus routing information, selective forwarding, sink-holes, wormholes, HELLO floods
Clustering based protocols (LEACH, TEEN, PEGASIS)	Selective forwarding, HELLO floods
Rumor routing	Bogus routing information, selective forwarding, sink-holes, Sybil, wormholes
Energy conserving topology maintenance (SPAN, GAF, CEC, AFECA)	Bogus routing information, Sybil, HELLO floods

Implementation:-

NS (Version 2) is an open source network simulation tool. It is an object oriented, discrete event driven simulator written in C++ and Otcl. The primary use of NS is in network researches to simulate various types of wired/wireless local and wide area networks; to implement network protocols such as TCP and UPD, traffic source behavior such as FTP, Telnet, Web, CBR and VBR, router queue management mechanism such as Drop Tail, RED and CBQ, routing algorithms such as Dijkstra, and many more.

Ns2 is written in C++ and Otcl to separate the control and data path implementations. The simulator supports a class hierarchy in C++ (the compiled hierarchy) and a corresponding hierarchy within the Otcl interpreter (interpreted hierarchy).

The reason why ns2 uses two languages is that different tasks have different requirements: For example simulation of protocols requires efficient manipulation of bytes and packet headers making the run-time speed very important. On the other hand, in network studies where the aim is to vary some parameters and to quickly examine a number of scenarios the time to change the model and run it again is more important.

In ns2, C++ is used for detailed protocol implementation and in general for such cases where every packet of a flow has to be processed. For instance, if you want to implement a new queuing discipline, then C++ is the language of choice. Otcl, on the other hand, is suitable for configuration and setup. Otcl runs quite slowly, but it can be changed very quickly making the construction of simulations easier. In ns2, the compiled C++ objects can be made available to the Otcl interpreter. In this way, the ready-made C++ objects can be controlled from the OTcl level.

Otcl basics

This chapter introduces the syntax and some basic commands of the Otcl language that are used by ns2. It is important to understand how Otcl works before moving to the part that deals with the creation of the actual simulation scenario.

Assigning values to variables

In tcl, values can be stored to variables and these values can be further used in commands:

```
set a 5
set b [expr $a/5]
```

In the first line, the variable a is assigned the value “5”. In the second line, the result of the command [expr $a/5], which equals 1, is then used as an argument to another command, which in turn assigns a value to the variable b. The “$” sign is used to obtain a value contained in a variable and square brackets are an indication of a command substitution.

Procedures

One can define new procedures with the proc command. The first argument to proc is the name of the procedure and the second argument contains the list of the argument names to that procedure. For instance a procedure that calculates the sum of two numbers can be defined as follows:

```
proc sum {a b} {
        expr $a + $b
}
```

The next procedure calculates the factorial of a number:

```
proc factorial a {
        if{$a<=1}{
            return 1
        }
    #here the procedure is called again
    expr $x*[factorial[expr $x-1]]
}
```

It is also possible to give an empty string as an argument list. However, in this case the variables that are used by the procedure have to be defined as global. For instance:

```
proc sum {} {
    global a b
    expr $a + $b
}
```

Files and lists

In tcl, a file can be opened for reading with the command:

```
set testfile [open test.dat r]
```

The first line of the file can be stored to a list with a command:

```
gets $testfile list
```

Now it is possible to obtain the elements of the list with commands (numbering of elements starts from 0) :

```
set first [lindex $list 0]
set second [lindex $list 1]
```

Similarly, a file can be written with a puts command:

```
set testfile [open test.dat w]
puts $testfile “testi”
```

Calling subprocesses

The command exec creates a subprocess and waits for it to complete. The use of exec is similar to giving a command line to a shell program. For instance, to remove a file:

```
exec rm $testfile
```

The exec command is particularly useful when one wants to call a tcl-script from within another tclscript. For instance, in order to run the tcl-script example.tcl multiple times with the value of the parameter “test” ranging from 1 to 10, one can type the following lines to another tcl-script:

```
for {set ind 1} {$ind <= 10} {incr ind} {
    set test $ind
    exec ns example.tcl test
}
```

Creating the topology

To be able to run a simulation scenario, a network topology must first be created. In ns2, the topology consists of a collection of nodes and links.

Before the topology can be set up, a new simulator object must be created at the beginning of the script with the command:

```
set ns [new Simulator]
```

The simulator object has member functions that enable creating the nodes and the links, connecting agents etc. All these basic functions can be found from the class Simulator. When using functions belonging to this class, the command begins with “$ns”, since ns was defined to be a handle to the Simulator object.

Nodes

New node objects can be created with the command:

```
setn0[$nsnode]
setn1[$nsnode]
setn2[$nsnode]
setn3[$nsnode]
```

The member function of the Simulator class, called “node” creates four nodes and assigns them to the handles n0, n1, n2 and n3. These handles can later be used when referring to the nodes. If the node is not a router but an end system, traffic agents (TCP, UDP etc.) and traffic sources (FTP,CBR etc.) must be set up, i.e, sources need to be attached to the agents and the agents to the nodes, respectively.

Agents, applications and traffic sources

The most common agents used in ns2 are UDP and TCP agents. In case of a TCP agent, several types are available. The most common agent types are:

- Agent/TCP – a Tahoe TCP sender
- Agent/TCP/Reno – a Reno TCP sender
- Agent/TCP/Sack1 – TCP with selective acknowledgement

The most common applications and traffic sources provided by ns2 are:

- Application/FTP – produces bulk data that TCP will send
- Application/Traffic/CBR – generates packets with a constant bit rate

- Application/Traffic/Exponential – during off-periods, no traffic is sent. During on-periods, packets are generated with a constant rate. The length of both on and off-periods is exponentially distributed.
- Application/Traffic/Trace – Traffic is generated from a trace file, where the sizes and interarrival times of the packets are defined.

In addition to these ready-made applications, it is possible to generate traffic by using the methods provided by the class Agent. For example, if one wants to send data over UDP, the method

```
send(int nbytes)
```

can be used at the tcl-level provided that the udp-agent is first configured and attached to some node.

Below is a complete example of how to create a CBR traffic source using UDP as transport protocol and attach it to node n0:

```
set udp0 [new Agent/UDP]
$ns attach-agent $n0 $udp0
set cbr0 [new Application/Traffic/CBR]
$cbr0 attach-agent $udp0
$cbr0 set packet_size_ 1000
$udp0 set packet_size_ 1000
$cbr0 set rate_ 1000000
```

An FTP application using TCP as a transport protocol can be created and attached to node n1 in much the same way:

```
set tcp1 [new Agent/TCP]
$ns attach-agent $n1 $tcp1 set
ftp1 [new Application/FTP]
$ftp1 attach-agent $tcp1
$tcp1 set packet_size_ 1000
```

The UDP and TCP classes are both child-classes of the class Agent. With the expressions [new Agent/TCP] and [new Agent/UDP] the properties of these classes can be combined to the new objects udp0 and tcp1. These objects are then attached to nodes n0 and n1. Next, the application is defined and attached to the transport protocol. Finally, the configuration parameters of the traffic source are set. In case of CBR, the traffic can be defined by parameters rate_ (or equivalently interval_, determining the interarrival time of the packets), packetSize_ and random_ . With the random_ parameter it is possible to add some randomness in the interarrival times of the packets. The default value is 0, meaning that no randomness is added.

Controlling the simulation

After the simulation topology is created, agents are configured etc., the start and stop of the simulation and other events have to be scheduled. The simulation can be started and stopped with the commands

```
$ns at $simtime “finish”
$ns run
```

The first command schedules the procedure finish at the end of the simulation, and the second command actually starts the simulation. The finish procedure has to be defined to flush the trace buffer, close the trace files and terminate the program with the exit routine. It can optionally start NAM (a graphical network animator), post process information and plot this information.
The finish procedure has to contain at least the following elements:

```
proc finish {} {
    global ns trace_all
    $ns flush-trace
    close $trace_all
    exit 0
}
```

Other events, such as the starting or stopping times of the clients can be scheduled in the following way:

```
$ns at 0.0 “cbr0 start”
$ns at 50.0 “ftp1 start”
$ns at $simtime “cbr0 stop”
$ns at $simtime “ftp1 stop”
```

If you have defined your own procedures, you can also schedule the procedure to start for example every 5 seconds in the following way:

```
proc example {} {
    global ns
    set interval 5
    ….
    …
    $ns at [expr $now + $interval] “example”
}
```

Simple Simulation Example

This section presents a simple NS simulation script and explains what each line does. Example 3 is an OTcl script which creates the simple network configuration and runs the simulation scenario in Figure 4. To run this simulation, copy the code in Example 3 in a file named "ns-simple.tcl" and type "ns ns-simple.tcl" at shell prompt.

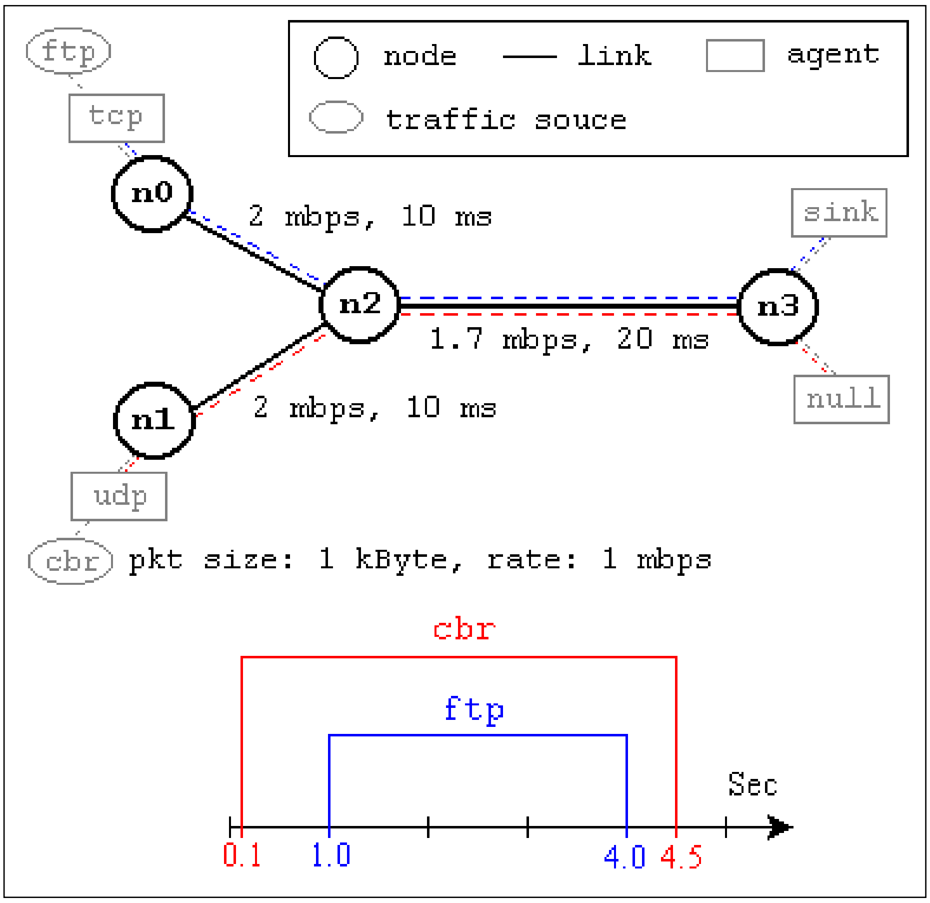
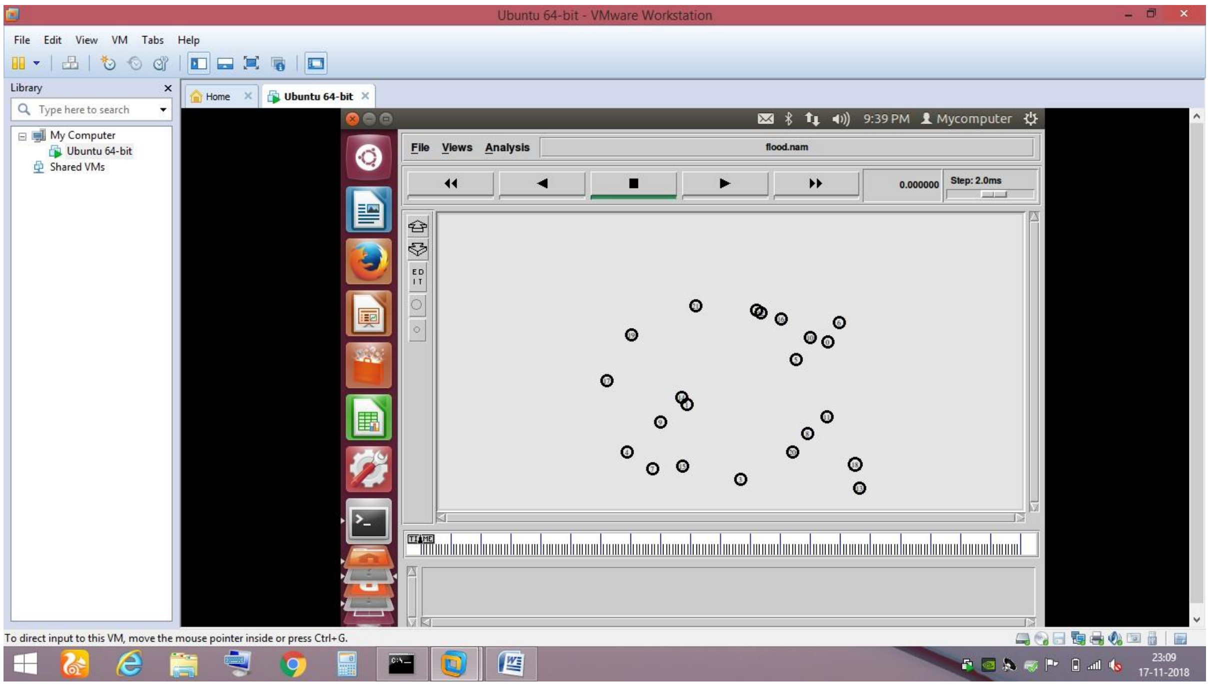
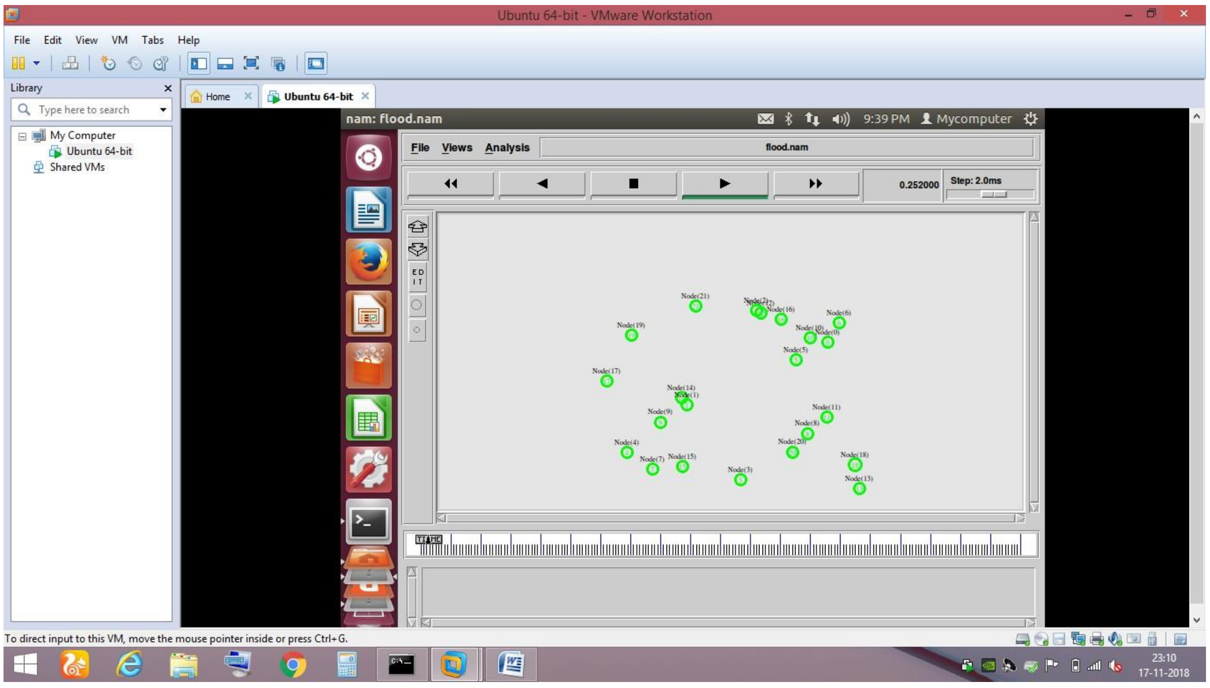


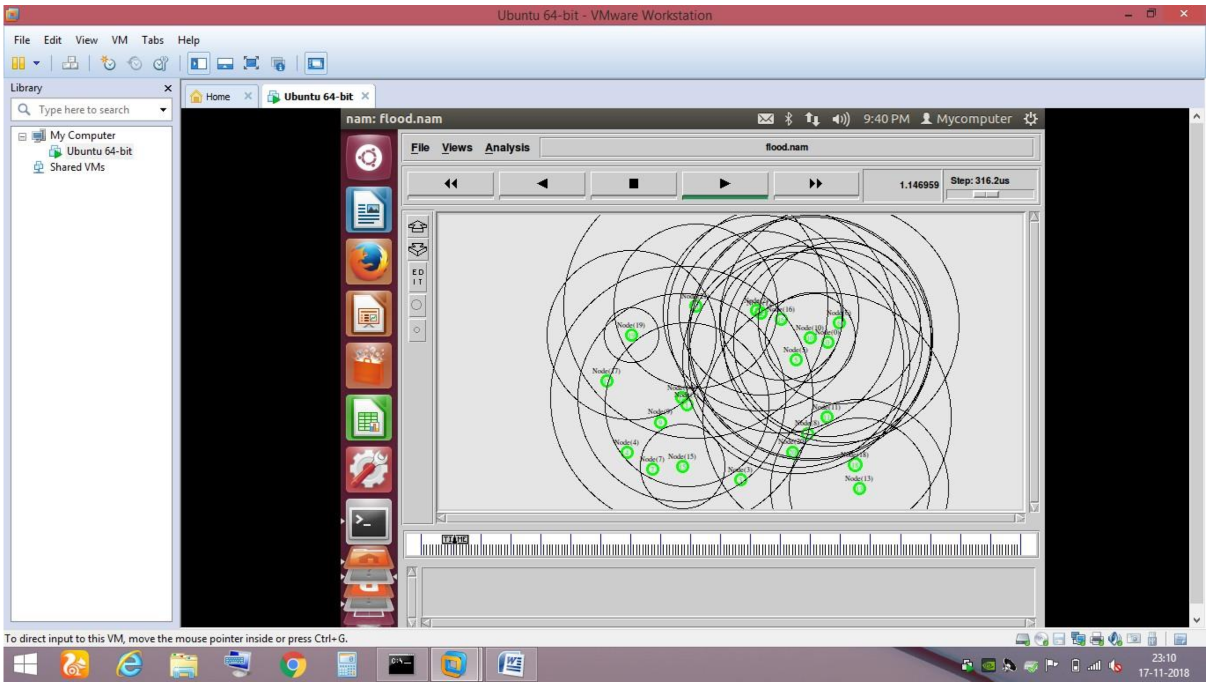
Figure 4. A Simple Network Topology and Simulation Scenario

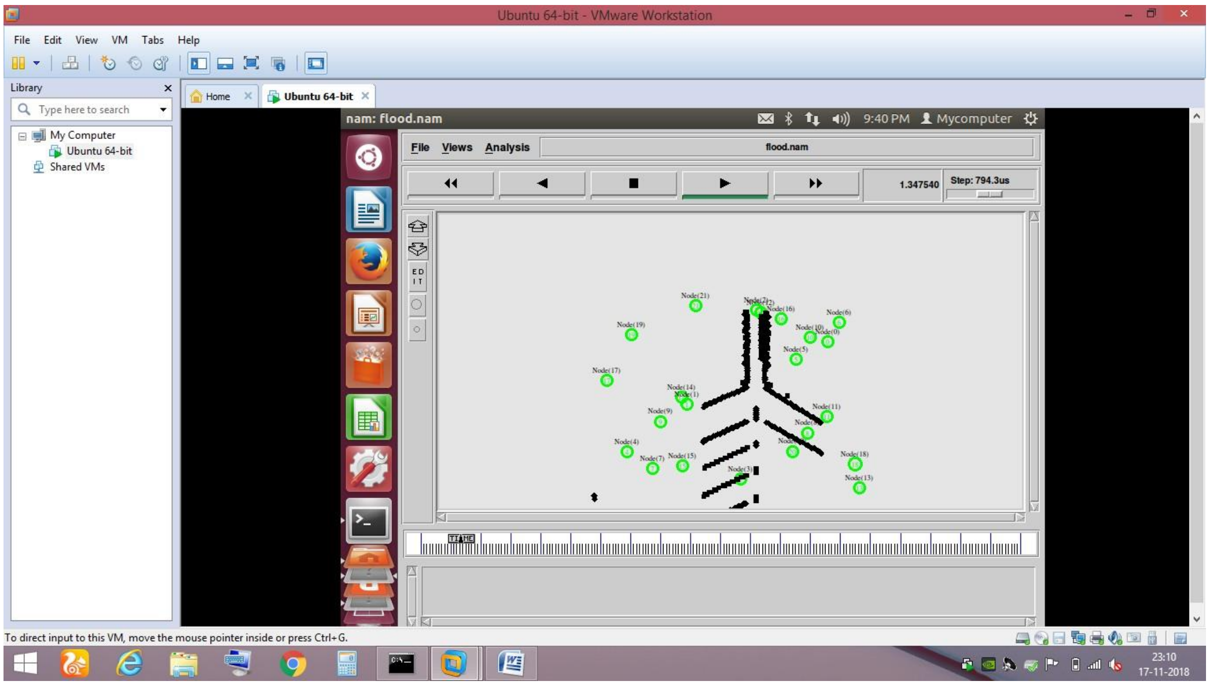
The network consists of 4 nodes (n0, n1, n2, n3) as shown in above figure. The duplex links between n0 and n2, and n1 and n2 have 2 Mbps of bandwidth and 10 ms of delay. The duplex link between n2 and n3 has 1.7 Mbps of bandwidth and 20 ms of delay. Each node uses a DropTail queue, of which the maximum size is 10. A "tcp" agent is attached to n0, and a connection is established to a tcp "sink" agent attached to n3. As default, the maximum size of a packet that a "tcp" agent can generate is 1KByte. A tcp "sink" agent generates and sends ACK packets to the sender (tcp agent) and frees the received packets. A "udp" agent that is attached to n1 is connected to a "null" agent attached to n3. A "null" agent just frees the packets received. A "ftp" and a "cbr" traffic generator are attached to "tcp" and "udp" agents respectively, and the "cbr" is configured to generate 1 KByte packets at the rate of 1 Mbps. The "cbr" is set to start at 0.1 sec and stop at 4.5 sec, and "ftp" is set to start at 1.0 sec and stop at 4.0 sec.

DISCUSSION OF RESULTS






Ubuntu 64-bit - VMware Workstation
nam: flood.nam
File Views Analysis
flood.nam
1.146959
Step: 316.2us
To direct input to this VM, move the mouse pointer inside or press Ctrl+G.


Ubuntu 64-bit - VMware Workstation
nam: flood.nam
File Views Analysis
flood.nam
1.347540
Step: 794.3us
To direct input to this VM, move the mouse pointer inside or press Ctrl+G.

CHAPTER -5

CONCLUSION

The wireless sensor networks continue to grow and become widely used in many mission-critical applications. So, the need for security becomes vital. However, the wireless sensor network suffers from many constraints such as limited energy, processing capability, and storage capacity, as well as unreliable communication and unattended operation, etc. There are many ways to provides security, In this paper we have discussed the various types of routing protocols because routing protocols plays a major role in wireless sensor networks. In which Ad-hoc Distance vector routing protocol plays a major role. We have presented the design and implementation of various attacks faced by AODV. The attacks are Sink hole, Grey hole, warm hole and sybil attack. The counter measures for this routing protocol is Ad-hoc On-demand Multipath Distance (AMODV). Because when compared to AODV the attacks faced by AMODV are less.so in future we will overcome the attacks by using AMODV routing protocols.

CHAPTER-6

REFERENCES

[1] Altman, E.; Jiménez, T. (2003). *NS Simulator for beginners* [Online]. Available: citeseer.ist.psu.edu/altman03ns.html

[2] *The ns Manual (formerly ns Notes and Documentation)* [Online]. Available: http://www.isi.edu/nsnam/ns/doc/index.html

[3] Chung, J.; Claypool, M. *NS by Example* [Online]. Available: http://nile.wpi.edu/NS/

[4] Greis, Marc. *Tutorial for the Network Simulator "ns"* [Online]. Available: http://www.isi.edu/nsnam/ns/tutorial/

[5] David martins and Herve Guyennet *IEEE2010 Wireless sensor networks attacks and security mechanisms*

[6] Khawla Naji Shnaikat1 and Ayman Ahmed AlQudah 2 key management schemes in wireless sensor networks