Detecting Attacks in Wireless Mesh Networks

Project Report Submitted in partial fulfillment of the requirement

For the degree of

Bachelor of Technology

In

Computer Science & Engineering

Under the supervision of

Dr. Hemraj Saini

By

Sakshi Rana (111239)

To

Jaypee University Of Information and Technology

Waknaghat, Solan-173234, Himachal Pradesh

Certificate

This is to certify that the project report entitled "**Detecting and Mitigating attacks in wireless mesh networks**", submitted by Sakshi Rana in partial fulfillment for the award of degree of Bachelor of Technology in Computer Science & Engineering to Jaypee University of Information and Technology, Waknaghat, Solan has been carried out under my supervision.

This work has not been submitted partially or fully to any other University or Institute for the award of this or any other degree or diploma.

Date: 15/5/2015 **Dr. Hemraj Saini**

Assistant Professor(Senior Grade)

Acknowledgement

First of all I wish to express my sincere thanks to **Prof. Dr. RMK Sinha**, Dean (CSE and IT), for providing me with all the necessary facilities.

I place on record, my sincere gratitude to Prof. Dr. **Satya Prakash Ghrera,** FBCS, SMIEEE Professor, Brig (Retd.) and Head, Dept. of CSE, for his constant encouragement.

I express profound gratitude to my guide **Dr. Hemraj Saini**, Assistant Professor (Senior Grade), Dept of CSE for his invaluable support, encouragement, supervision and useful suggestions throughout this project work.

I take this opportunity to record our sincere thanks to all the faculty members of Department of Computer Science and Engineering, for their help and encouragement. I also thank my parents for their unceasing encouragement and support.

I am thankful and indebted to all those who helped me directly or indirectly in completion of my work.

Date: 15/5/2015 **Sakshi Rana**

Table of Content

Abbreviations and Symbols

S.No.	Abbreviation	Stands for
1	WMN	Wireless Mesh Network
2	MP	Mesh Point
3	MAP	Mesh Access Point
4	MANET	Mobile Ad-Hoc Network
5	DoS	Denial-of-Service
6	SS	Spread-Spectrum
7	PN	Pseudo-random Noise
8	ACK	Acknowledgement

S.No.	Symbol	Stands for
1	$\in$	Belongs to
2	$\rightarrow$	Maps on
3	$\|x\|$	Length of x
4	$\|\|$	Concatenate
5	$\oplus$	EX-OR

List of Figures

List of Tables

Abstract

With the advent of various types of wireless networks in the next generation to provide users with better services and faster technology, wireless mesh networks have been gaining immense popularity. With their unique architectural features and fast connectivity wireless mesh networks promise to extend performance beyond what is obtained with the current WI-FI based infrastructure. Wireless mesh networks not only provide users with unmatched connectivity and performance but also has features like dynamically self-organized and self-configuring. This feature brings many advantages to WMNs such as low-upfront cost, easy network maintenance, robustness and reliable service coverage. Despite all the benefits, wireless mesh networks are highly susceptible to internal as well as external attacks due to the open nature of the wireless medium. Although external attacks can be mitigated using a combination of cryptography based and robust communication techniques, the internal attacks, which exploit the knowledge of the network secrets and protocol semantics, are difficult to detect and require protocols with built-in security measures. Through this project I try to address the problem of Jamming attacks in wireless mesh networks. In these type of attacks the attacker takes advantage of the network secrets and then classifies the packets which are of high importance .To perform Selective jamming, the attacker must be capable of classifying the transmitted packets and corrupting them before the end of their transmission. Through this project I try to detect the various kinds of security threats that hinder the performance of wireless mesh networks. Also, try to formulate new strategies and also improve existing ones to mitigate these attacks on WMNs. Furthermore I am interested in identifying the loopholes of the existing strategies to make them more robust and make them efficient to detect and mitigate attacks more effectively. Hence, to provide a reliable communication network.

CHAPTER1. INTRODUCTION

Wireless mesh networks (WMNs) consist of mesh routers and mesh clients, where mesh routers have minimal mobility and form the backbone of WMNs. Such networks evolve from classic mobile ad hoc networks (MANETs).In simpler terms we can say that WMN is a particular type of MANET which aims to provide ubiquitous high bandwidth access for large no. of users. WMNs follow a two tier architecture, wherein the first tier consists of the users (i.e. Stations) and the second tier consists of peer-to-peer network between the mesh access points (MAPs) (Fig. 1.1). Each node in a WMN operates not only as a host but also as a router, forwarding packets on behalf of other nodes that may not be within direct wireless transmission range of their destination. A WMN is dynamically self configuring and self healing network. WMNs are inherently designed to be more robust and resilient .Wireless Mesh Networks are ideal in scenarios where wiring is inconvenient.A mesh infrastructure brings Internet access to low-income places where telecom operators don't have access to.

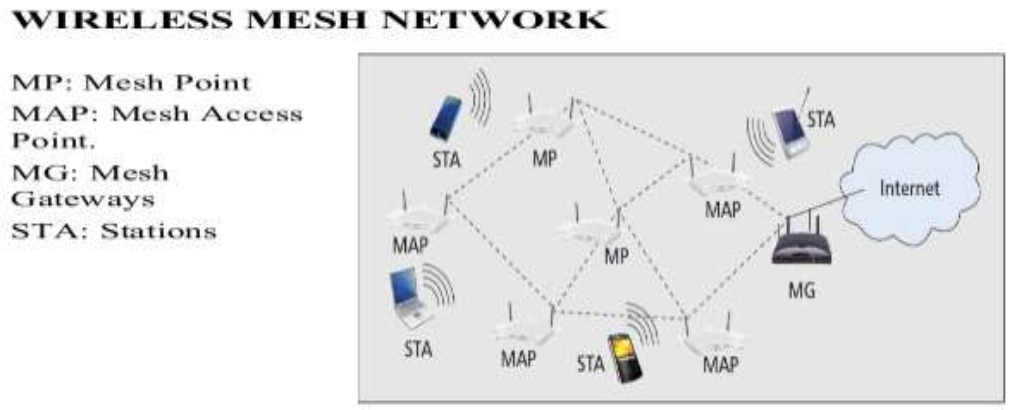


Fig1.1 WMN Architecture

Wireless mesh networking is a promising wireless technology for a number of applications, like, broadband home networking, community and neighborhood networks, enterprise networking etc. It is gaining popularity as a possible way to roll out robust and reliable wireless broadband service access in a way that needs minimal up-front

investment. Also with the capability of self-healing and self-configuring, WMNs can be deployed as per the need, one node at a time.

1.1 Network Architecture

In most general form, a wireless mesh network interconnects stationary and/or mobile clients and optionally provides access to the internet. The defining characteristic of WMNs is that each node of the network forward packets on behalf of the nodes not in the range of wireless transmission.

On the basis of the functionality of the nodes the architecture of WMNs can be classified into three main categories i.e. Backbone WMNs, Client WMNs and Hybrid WMNs.

- Infrastructure/Backbone WMNs: The Infrastructure/Backbone WMN consists of mesh routers, which form an infrastructure for clients that connect to them. The mesh routers in this architecture form a backbone of self healing, self configuring links among them. The architecture is shown in Figure 1.2 where dashed and solid line indicate wireless and wired links, respectively.

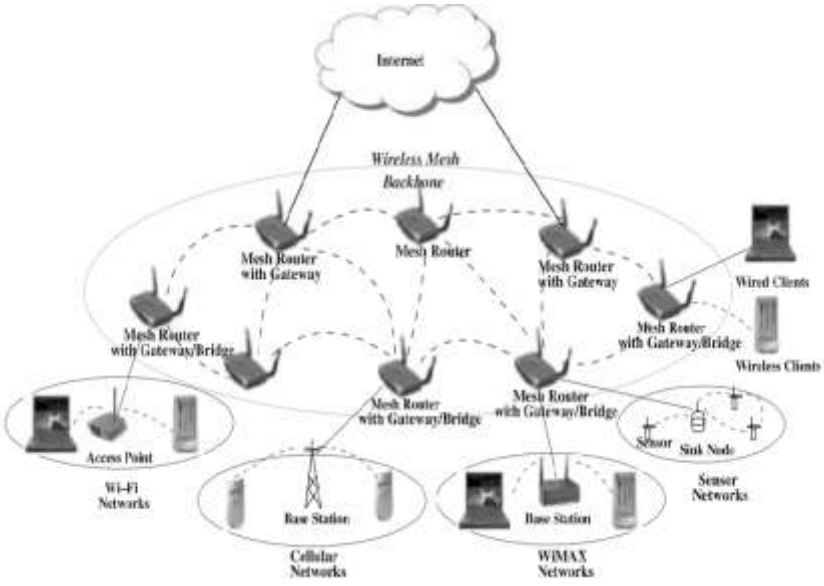


Fig 1.2 Infrastructure WMNs

- Client WMNs: This type of architecture provides peer-to-peer networks among the devices. In client WMNs, client nodes constitute the actual network to perform routing and configuration functionalities as well as

providing end-user applications to the customers. The basic architecture is shown in Figure 1.3. In client WMNs, the end users have to perform additional functions such as routing and self-configuration.

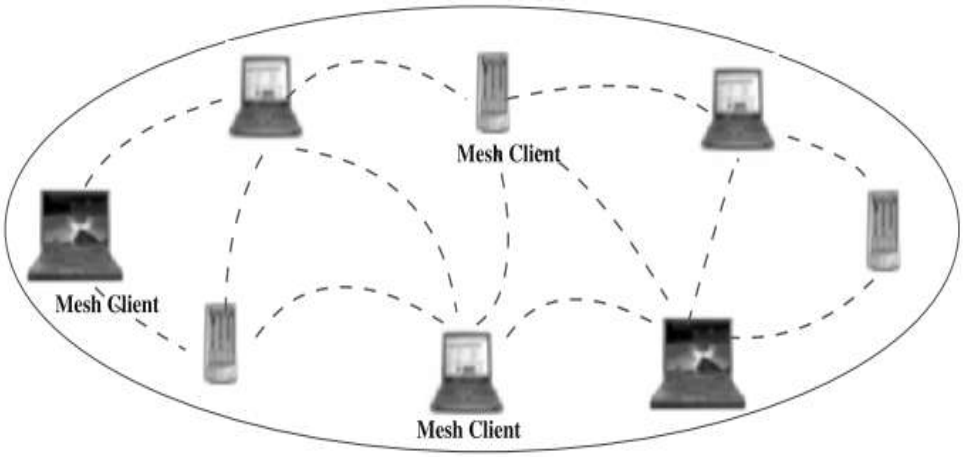


Fig 1.3 Client WMNs

- Hybrid WMNs: This type of architecture can be said to be the combination of both infrastructure and client WMN. On the one hand where the infrastructure provides connectivity to the other networks ,on the other hand mesh clients can access the network through mesh routers as well as directly meshing with other mesh clients. The architecture is shown in figure 1.3.

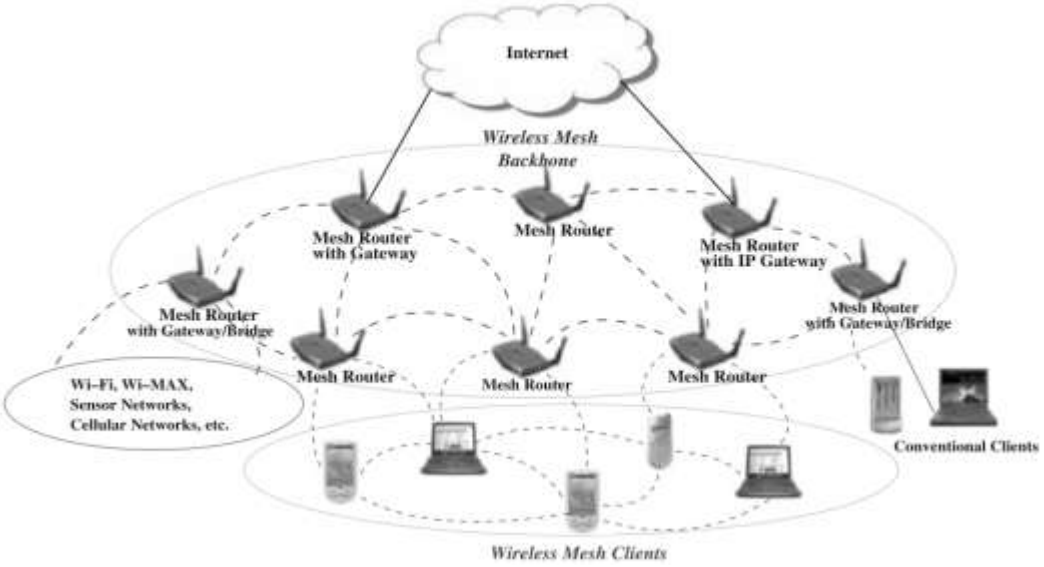


Fig 1.4 Hybrid WMN

1.2 Characteristics of WMN

- Low investment: Since the network avoids the cost of cables, the installation cost is very low. Also the network enables growth to be gradual, as per the need.

- Support for ad hoc networking, and capability of self-forming, self-healing, and self-organization: Ad hoc networking enhances network performance, such as flexible network architecture, easy deployment and configuration, fault tolerance, and mesh connectivity, i.e., multipoint-to-multipoint communications.

- Customer Coverage: Due to its multi hop routing ability, line of sight to a single base station is not required; as long as a client has connectivity to another client, it can obtain Internet access.

- Network access to different types: WMNs support both, access to internet and peer-to-peer communication within the WMN. Also, integration of WMNs with other wireless networks and providing services to end-users of these networks can be accomplished through WMNs.

- Mobility: The infrastructure of WMN is provided by the mesh routers hence the coverage of the network can be engineered easily. With WMNs it is possible to provide continuous connectivity throughout the network, without compromising the performance, while still supporting the mobility of end users.

- Compatibility: WMNs are compatible with previous technologies. The existing techniques developed for ad hoc networks are already applicable to WMNs.

- Wireless infrastructure/backbone: The wireless backbone provides wide coverage, connectivity, and robustness in the wireless domain.

- Wireless Mesh Networks are a wireless multi-hop networks in which the nodes are characterized by theirs stability. In this sort of network the minimum cost tree connects sources and receivers by implying a minimum number of forwarding nodes.

Sno.	Pros	Cons
1.	They are self healing: if any node fails, another will take place.	They're still in development.
2.	As the network gets bigger, it also gets faster.	New standards have not yet been adopted.
3.	They're useful where line-of-sight wireless signals are intermittently blocked.	Wireless links are inherently unreliable. Since this problem gets worse with each hop, the size of meshes is currently limited.
4.	LANs can run faster than other networks because local packets don't need to run back to a central server.	They're not completely seamless. Moving nodes (e.g., those in vehicles) may not establish new connections easily. When a network's topology changes, some transmission paths can be temporarily disrupted. Thus, voice and video don't work as well on meshes.

Table Pros and Cons of WMNs

1.3Vulnerabilities of WMN

All types of wireless networks are susceptible of numerous security threats but Wireless Mesh Networks are particularly vulnerable to them, for a number of reasons.

- First, MPs and MAPs are relatively cheap devices with poor physical security, which makes them potential, targets for node capture and compromise.

- Second, given their relatively advanced hardware (e.g., multiple ktransceivers per MP and MAP), WMNs often adopt a multi-channel design, with one or more channels dedicated for control/broadcast purposes. Such static design makes it easier for an attacker to selectively target control/broadcast information.

- Third, the reliance on multi hop routes further accentuates the WMN vulnerability to compromised relays which can drop control messages, in order to enforce a certain routing behavior (e.g., force packets to follow long or inconsistent routes).

These factors leave WMNs vulnerable to various attacks which can lead to compromise of user's privacy. These attacks exploits knowledge of network secrets and protocol semantics to attack critical network functions such as channel access, routing, and end-to-end reliable data delivery.

We, therefore, need to come up with strategies to mitigate these attacks in order to provide users with a reliable connection.

In the next section we discuss the different kind of attacks that can be launched to hamper a WMN.

CHAPTER 2. ATTACKS IN WMNs

An attack can be defined as any kind of malicious activity that attempts to collect, disrupt, deny, degrade or destroy information system resources or the information itself. An attack can be either active or passive. In an active attack the attacker will attempt to alter the system resources or affect their operation in some way. But in case of passive attacks the attacker does not affect the system resources, instead tries to gain information. In passive attacks the attacker monitors unencrypted traffic or looks for clear text passwords and sensitive info that can be later used in some other attack. Passive attacks basically mean that the attacker is eavesdropping.

The attack can be perpetrated internally or externally. The external attacks are the ones launched by "foreign" devices that are unaware of the network secrets. They can take the forms of random channel jamming, packet relay and packet fabrication.

The internal attacks, however, are launched by exploiting the knowledge of network secrets and are difficult to mitigate. They require protocols with built-in security measures, through which the attacker can be detected and its selective nature can be neutralized.

One of the fundamental ways of degrading the network performance for a WMN is by Jamming. In the simplest form of jamming, the adversary corrupts transmitted messages by causing electromagnetic interference in the network's operational frequencies, and in the proximity to the targeted receivers. This is known as "always-on" jamming strategy. Jammers can be categorized into four models:

- **Constant Jammer**: continuously emits noise.
- **Deceptive Jammer**: continuously broadcasts fabricated messages / replay old ones.
- **Random Jammer**: alternates between period of jamming and inactivity.
- **Reactive Jammer**: jams only when transmission is detected.

Consider the scenario depicted in figure 2.1. Nodes A and B are communicating over a wireless medium and jamming node J is within the communication range of both A and B. Node A transmits a packet to node B which is being eavesdropped by node J. Node J classified the packet being transmitted and corrupts it by interfering with its reception at B.

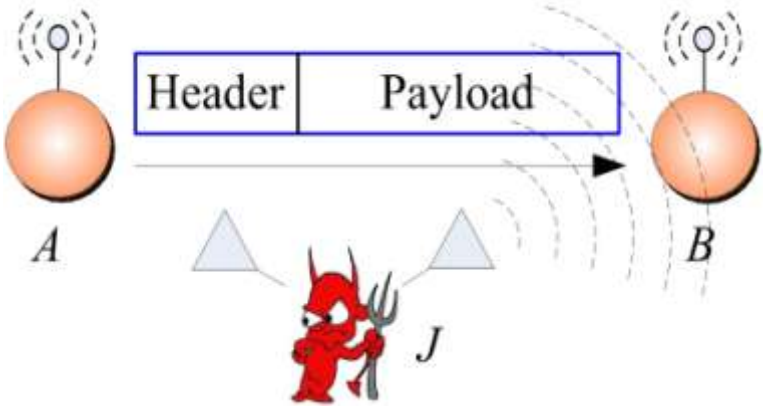


Figure 2.1 Realization of a selective Jamming Attack

The types of attacks that can be selectively launched on a WMN are:

- **Selective Jamming Attacks**
- **Selective Dropping Attacks**

2.1 Selective Jamming Attacks

Jamming attacks in wireless medium has been primarily analyzed under an external adversarial model, as a severe form of denial of service (DoS) against the physical layer. The strategies that exist to mitigate these attacks involve the use of some form of spread-spectrum (SS) communication in which the signal is spread across a large bandwidth on the basis of pseudo-noise (PN) code. However SS can protect wireless communication only to the extent that PN code remains a secret. Insiders with the knowledge of PN code can still launch jamming attacks. Selective Jamming attacks against WMNs can employ either channel selectivity or data selectivity.

Channel Selective Jamming: In a typical WMN, one or more channels are reserved for broadcasting control information. These channels, referred to as control channels, facilitate operations such as network discovery, time synchronization, coordination of shared medium access, and routing path discovery. An adversary who selectively targets the control channels can efficiently launch a DoS attack with a fairly limited amount of resources (control traffic is low-rate compared to data traffic). To launch a channel-selective jamming attack, the adversary must be aware of the location of the targeted channel, whether defined by a separate frequency band, time slot, or PN code.

Data Selective Jamming: To further improve the energy efficiency of selective jamming and reduce the risk of detection, an inside attacker can exercise a greater degree of selectivity by targeting specific packets of high importance. One way to launch a data-selective jamming attack is by classifying packets before their transmission is completed.

2.2 Selective Dropping Attacks

If selective jamming is not successful due to anti-jamming measures, an insider can selectively drop packets post-reception.

Once a packet has been received, the compromised node can inspect the packet headers, classify the packet, and decide whether to forward it or not. Such an action is often termed misbehavior.

Post-reception dropping is less flexible than selective jamming because the adversary is restricted to dropping only the packets routed through it. Nonetheless, the impact on the WMN performance can be significant.

Figure 2.2 depicts how a compromised MP throttles the rate of end-to-end connection by selectively dropping the critical control packets.

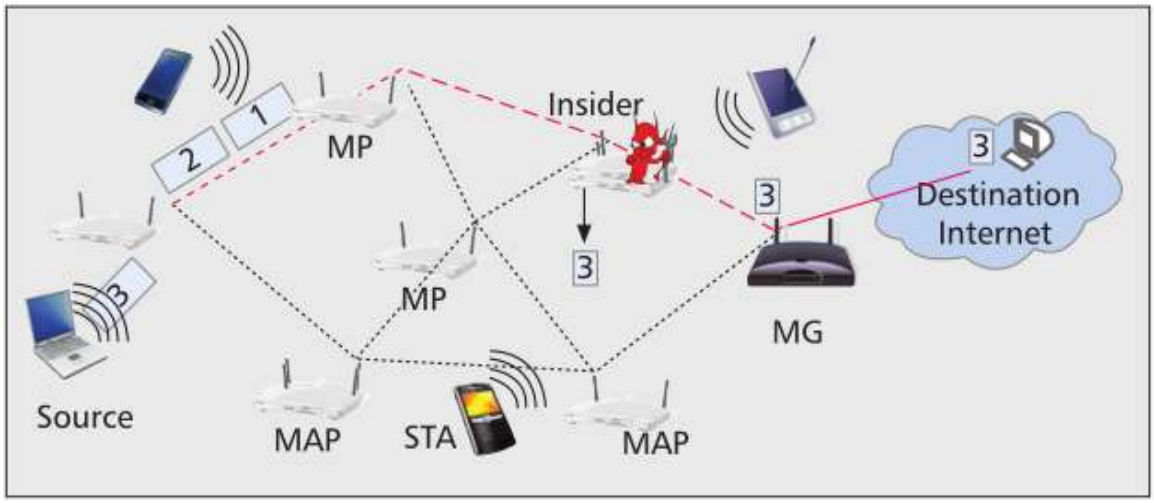


Figure 2.2 An insider selectively drops the cumulative TCP acknowledgements and forces end-to-end data retransmission

The dropping of cumulative TCP acknowledgments results in the end-to-end retransmission of the entire batch of pending data packets (Fig. 4).

In addition, packet loss is interpreted as congestion, resulting in the throttling of the sender's transmission rate.

In this project we have focused on selective jamming attacks.

CHAPTER 3 MITIGATION TECHNIQUES FOR SELECVTIVE JAMMING

In this section, we propose possible schemes for countering selective jamming. Our goal is to transform a selective jammer to a random one. By transforming a selective jammer into a random one, we can then apply the algorithms defined. As selective jamming attacks are in multi hop transmission and we have no idea about when they will occur, hence they are very difficult to mitigate. But if we convert it into random jammer we can formulate algorithms into removing the selective jamming attacks.

A. Scheme based on Commitments

In cryptography, a **commitment scheme** allows one to commit to a chosen value (or chosen statement) while keeping it hidden to others, with the ability to reveal the committed value later.

Commitment schemes are designed so that a party cannot change the value or statement after they have committed to it: i.e. commitment schemes are *binding*. Commitment schemes have important applications in a number of cryptographic protocols.

Interactions in a commitment scheme take place in two phases:

1. the *commit phase* during which a value is chosen and specified
2. the *reveal phase* during which the value is revealed and checked

The scheme being discussed can be termed as Strong Hiding Commitment Scheme.

In this scheme we have a sender S which performs role of the committer too. Say we have to send a packet message m $\in \{0,1\}^l$, where l is the length of the packet.

First, we select a random key k $\in \{0,1\}^q$, where q is the number of bits mapped to a symbol at the physical layer. To utilize the off-the-shelf encryption algorithm, k is expanded to k_1=f (k), where f : $\{0,1\}^q \rightarrow \{0,1\}^z$ is a public injective function and z=|k_1| is the length required in block encryption mechanism such as DES or AES. After generation

of k_1, S generates the committed value C=E_{k_1}(m) and broadcasts {C, h_{k_1}(m||k)}, where h_{k_1} is a collision-resistant keyed one-way hash function.

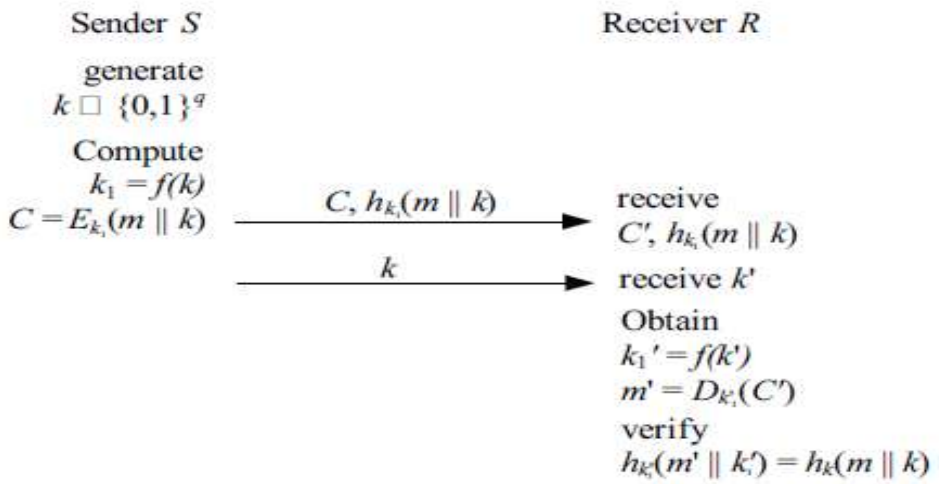


Figure 3.1 A commitment scheme for preventing packet classification

Now to "open" C, S releases the random key. Upon reception of a k', now R computes k'_1=f (k'), and obtains m'= $D_{k'_1}$(C').

The integrity of the message (i.e. m'=m) is verified by checking $h_{k'_1}$(m'||k') = h_{k_1}(m||k). Upon verification, R obtains m'=m. The same procedure is represented in figure 3.1.

To classify m, the jammer must be capable of obtaining any part of m before the end of the transmission of k.

The communication overhead introduced by the commitment scheme is equal to the lengths of h_{k_1}(m||k) and k (i.e160 + q bits) per packet.

To improve the efficiency we can use the same key to encrypt n consecutive data messages.

B. A Scheme based on Cryptographic Puzzles

Cryptographic puzzles involve the creation of problems that are solvable within a finite time interval t_p which depends on the hardness of the puzzle and the computational ability of the solver.

Such puzzles were first suggested by Merkle and have found various applications including the prevention of DoS attacks. In our project, the idea of cryptographic puzzles can be employed to overwhelm the computational ability of the adversary in classifying packets.

In essence, this can be an implementation of a commitment scheme where the committer never reveals the information needed to open the commitment, but such information is obtained after solving a puzzle. The time required for solving a puzzle can be controlled by using time-lock puzzles.

Assume that S wants to broadcast a message m $\in \{0,1\}^l$. S generates a composite modulus n = p × q where p and q are two large random prime numbers. Then he computes φ (n) = (p−1) (q−1) and t=D×T, where D is the number of squarings modulo n per second that a device can perform, and T is the time that it takes to solve the puzzle. S encrypts m with a randomly selected key, k $\in \{0,1\}^s$, using a conventional symmetric algorithm such as AES, getting E_k(m).

Then S chooses a random number a modulo n and computes C_k= k +a^{2t} (mod n), that could be done efficiently by defining e =2^t (mod φ (n)) and computing C_k = k + a^e (mod n). Finally, S transmits {n, a, t, E_k (m), C_k }.

Receiver R has to compute b = a^{2t} (mod n) to get k, and then m. Note that, without knowing p and q there is no efficient alternative to get b, but to perform the necessary squaring operations.

A value of T equal to the transmission delay of C_k is sufficient to prevent the disclosure of any part of m before of the end of the transmission of C_k.

In this scheme, the transmission of {n, a, t, C_k} introduces a communication overhead of 40 bytes per packet (n, a, t are double numbers of 8 bytes, and C_k is a key of size 128 bits).

As in the case of commitment schemes, to improve the efficiency of this scheme, we can use the same key to encrypt n consecutive data messages. In this case, the receiver only solves the puzzle once for every n data packets transmitted, and the delay is reduced by (n−1) ×T amount of time.

If the computational power of the receiving node is a concern, computationally efficient cryptographic puzzles based on the partial reveal of the input to one-way hash functions can be used. However, such puzzles are pararellizable and can be solved in a shorter time from a computationally advanced adversary.

To prevent pararellization, puzzles can be constructed by the iterative application of an encryption function such as AES with partial reveal of the decryption key. In this case, the adversary has to perform a number of decryptions in a sequential manner.

C. A Scheme Based on All-Or-Nothing Transformations

In cryptography, an **all-or-nothing transform** (**AONT**), also known as an **all-or-nothing protocol**, is an encryption mode which allows the data to be understood only if all of it is known.

AONTs are not encryption, but frequently make use of symmetric ciphers and may be applied before encryption. In exact terms, "an AONT is an unkeyed, invertible, randomized transformation, with the property that it is hard to invert unless all of the output is known". NT). Such transformations were originally proposed by Rivest to slow down brute force search attacks.

An AONT serves as a publicly known and completely invertible pre-processing step to a plaintext, before it is passed to an ordinary block encryption algorithm.

The defining property of an AONT is that the entire output of the transformation must be known before the input can be computed.

When combined with block encryption, all blocks of the cipher text must be decrypted to obtain any part of the plaintext, thus slowing down a brute force attack by a factor equal to the number of cipher text blocks.

Following AONT transform known as the package transform was proposed by Rivest:

1. The input message m is split into blocks $m_1, m_2, m_3, \ldots\ldots\ldots\ldots, m_x$.
2. A random key k' $\in \{0,1\}^s$, is selected.
3. The output message m' consisting of $m'_1, m'_2, m'_3, \ldots\ldots\ldots\ldots, m'_x, m'_{x+1}$ is computed as follows:

   $m'_i = m_i \oplus E_{k'}$ (i), for i=1,2,……..,x,

   m'_x= k' $\oplus e_1 \oplus e_2 \oplus e_3 \oplus \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \oplus e_x$,

Where,

$e_i = E_{k0}(m'_i \oplus$ i), for i=1,2,……,x

And k_0 is a fixed publicly-known encryption key. In this transform, to obtain k', all m'_i must be known ,

k'= $m'_{x+1} \oplus e_1 \oplus e_2 \oplus e_3 \oplus \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \oplus e_x$,

Once k' is known, we can obtain m_i's as follows:

$m_i = m'_i \oplus E_{k'}$ (i), for i=1,2,……..,x,

CHAPTER 4 MITIGATION TECHNIQUES FOR SELECVTIVE DROPPING

Selective dropping attacks can be mitigated by employing fault-tolerant mechanisms at various layers of the protocol stack.

At the routing layer, multi-path routing provides robust multi-hop communication in the presence of network faults, by utilizing more than one path from a source to a destination. At the transport layer, variants of the standardized TCP protocol have been specifically developed for dealing with the imperfections of the wireless medium. These protocols differentiate between congestion and wireless transmission losses.

A selective dropper can always attribute his losses to congestion, in order to avoid detection as a malicious node. In this case, identification mechanisms employing long-term statistics, can accurately pinpoint selective droppers**.**

The existing methods for detection of misbehavior in WMNs are reputation, credit-based and acknowledgement systems.

Reputation system: These systems identify the misbehaving node as per the node reputation metrics. This metrics is computed based on interactions of each node with its peers.

The two basic operations of reputation systems are: the collection of accurate observations of nodes' behavior and the computation of the reputation metric.

Behavioral information is collected based on first-hand observations provided by neighboring nodes and second-hand information provided by other interacting peers. Firsthand observations are collected by monitoring nodes that operate in promiscuous mode in order to verify the correct forwarding of transmitted packets.

Overhearing becomes problematic in the case of multichannel WMNs, because MPs and MAPs are scheduled to communicate in parallel over orthogonal frequency bands, and hence might not be available to monitor the behavior of other nodes.

Several schemes have been proposed for managing second-hand information. A node may flood warnings to the entire network if it detects a misbehaving node. Alternatively, information can be provided on demand after a request from a particular node has been received.

In the latter scenario flooding of the request is necessary to discover nodes that possess second-hand information. Both methods consume considerable bandwidth resources due to the underlying flooding operations for the dissemination and collection of second-hand information.

Robust computation of reputation metrics is equally important for the identification of packet droppers. Simple aggregate metrics have been shown to be vulnerable to false accusations from colluding malicious nodes and suddenly changing behavioral patterns. For instance, a misbehaving node can exhibit a long history of good behavior in order to build a high reputation metric before it starts to misbehave.

Such instances are dealt by assigning larger weights to recent behavioral observations and/or adopting additive increase multiplicative decrease type algorithms for updating the reputation metrics.

A critical challenge for any metric computation algorithm is the selective nature of packet droppers. When a very small fraction of packets is dropped, metrics that do not take into account the packet type are bound to have high rates of misdetection.

Dropping selectivity can be detected with the use of storage-efficient reports (e.g., based on Bloom filters) of the per-packet behavior of nodes.

Based on these reports, it is possible to conduct multiple tests to identify malicious selective dropping patterns.

These patterns are likely to have some deterministic structure compared to packet losses due to congestion or poor channel quality.

ACK-Based Systems: Acknowledgment (ACK)-based schemes differ from overhearing techniques in the method of collecting first-hand behavioral observations. Downstream nodes (more than a single hop away) are responsible for acknowledging the reception of messages to nodes several hops upstream.

These systems are suitable for monitoring the faithful relay of unicast traffic, at the expense of communication overhead for relaying an additional set of ACKs.

However, ACK-based schemes cannot be used to identify insiders that selectively drop broadcast packets. Such packets remain, in general, unacknowledged in wireless networks to avoid an ACK implosion situation.

Moreover, a small set of colluding nodes can still provide authentic ACKs to upstream nodes while dropping packets.

Credit-Based Systems: Credit-based systems alleviate selfish behavior by motivating nodes to forward packets. Nodes that relay traffic receive credit in return, which can be spent later to forward their own traffic.

However, in the context of WMNs, MPs do not generate any traffic of their own, but act as dedicated relays. Hence, compromised MPs have no incentive for collecting credit. Moreover, in the case of selective dropping attacks, misbehaving nodes can still collect sufficient credit by forwarding packets of low importance while dropping a few packets of high value.

In addition, the credit collected by a particular node depends on the topology of the network. A highly connected node is expected to collect more credit due to the increased volumes of traffic routed through it. An adversary compromising such a node is likely able to implement a selective dropping strategy without running out of credit.

Finally, credit-based systems lack a mechanism for identifying the misbehaving node(s), allowing them to remain within the network indefinitely. Selective dropping attacks are less flexible and more restricted as compared to Selective jamming attacks. As the attacker is limited to the packets that have been routed through. But still they can affect the performance and throughput of WMN significantly.

CHAPTER 5.

5.1 Literature Survey

The following research papers contributed to the understanding of the topic:

S.No	Title	Year
1.	Wireless Mesh Networks: Opportunities and challenges	2005
2.	Selective Jamming Attacks in Wireless Networks	2010
3.	Selective jamming/dropping insider attacks in wireless mesh networks	2011
4.	Delay minimization and priority scheduling in wireless mesh networks	2014

Wireless Mesh Networks: Opportunities and challenges

ML Sichitiu

Wireless World Congress, 2005 - ncsu.edu

Abstract

Wireless mesh networks have the potential to deliver Internet broadband access, wireless local area network coverage and network connectivity for stationary or mobile hosts at low costs both for network operators and customers. The core technology involves a network of wireless routers relaying each others' packets in a multihop fashion. Many variations on targeted applications and implementation choices offer different opportunities to emerging companies in this emerging area. In this article, the author presents an introduction to wireless mesh networks and present both the benefits enabled by this technology and the main hurdles that have to be overcome.

Selective Jamming Attacks in Wireless Networks

A Proano, L Lazos

Communications (ICC), 2010 IEEE

Abstract

The author addresses the problem of selective jamming attacks in wireless networks. In these attacks, the adversary selectively targets specific packets of “high” importance by exploiting his knowledge on the implementation details of network protocols at various layers of the protocol stack. They illustrate the impact of selective jamming on the network performance by illustrating various selective attacks against the TCP protocol. They show that such attacks can be launched by performing real-time packet classification at the physical layer. They examine the combination of cryptographic primitives with physical layer attributes for preventing real-time packet classification and neutralizing the inside knowledge of the attacker.

The author talks about how Wireless Mesh Networks are susceptible to numerous security threats due to the open nature of the wireless medium. They explain the various jamming strategies and categorize the ones they are going to work on. They in the paper consider a sophisticated adversary model in which the adversary is aware of the implementation details of the network protocol.

To mitigate the selective jamming attacks they have combined cryptographic mechanisms such as commitment schemes, cryptographic puzzles and all-in-one transformations with the physical layer parameters.

After the discussion of various mechanisms to mitigate the attack the author discusses the impact that selective jamming has on TCP. They discuss the results the recorded on OPNET modeler 14.5. Further they have done performance evaluation of the attacked network and the network with the mitigation techniques.

They analyze results using graphs which show jamming probability for delay, throughput and packets jammed.

Selective jamming/dropping insider attacks in wireless mesh networks

L Lazos, M Krunz

IEEE network, 2011

Abstract

Wireless mesh networks promise to extend high-speed wireless connectivity beyond what is possible with the current WiFi-based infrastructure. However, their unique architectural features leave them particularly vulnerable to security threats. In this article the author describes various forms of sophisticated attacks launched from adversaries with internal access to the WMN. They further identify possible detection and mitigation mechanisms.

The author talks of how Wireless mesh networks continue to receive significant nterest as a possible means of providing seamless data connectivity, especially in urban environments.

The author describes that WMN follows a two tier architecture where the first tier is the end user and the second tier is peer-to-peer network of MAPs.

Then the author discusses the vulnerabilities of WMNs and goes on to describing the Selective Jamming and Selective Dropping attacks. Then they discuss the mitigation techniques for both these attacks.

In the conclusion section the author writes how WMN is prone to external and internal attacks. They say that while most of the external attacks can be mitigated it's the internal attacks that pose a greater threat and are more harmful for the network.

Furthermore, they discuss a few challenges that need to be overcome. Like, jamming resistant broadcast communication in presence of insider jammers is posed as a challenge.

Further challenges include efficient behavioral monitoring mechanisms not relying on continuous overhearing of efficient maintenance and dissemination of reputation metrics.

Delay minimization and priority scheduling in wireless mesh networks

C Liu, B Fu, HJ Huang

Wireless Networks, 2014 – Springer

Abstract

Wireless mesh networks (WMNs) have emerged as a significant technology for applications because of its advantage of multi-radio and multi-channel which makes it perform better than wireless LANs. Furthermore, quality-of-service (QoS) support can be achieved by some distinguished ways in WMN. In this paper, QoS requirements are recorded by traffic profile, QoS constraints are formulated as delay time of transmitting all the requested data flows in the network. Multi-commodity flow technologies are applied for handling this issue. After minimizing the delay of the network by the assistance of multi-commodity-flow techniques and resource contention graph, we use effective channel assignment algorithm to schedule the data flows under the QoS constraints. Our evaluation indicates that our technologies successfully route flows under their special QoS requirements with different priority.

5.2 Software Approach

5.2.1 Ubuntu

Ubuntu is an ancient African word meaning ‘humanity to others’. It also means ‘I am what I am because of who we all are’. The Ubuntu operating system brings the spirit of Ubuntu to the world of computers.

The vision for Ubuntu is part social and part economic: free software, available to everybody on the same terms, and funded through a portfolio of services provided by Canonical.

Ubuntu is different from the commercial Linux offerings that preceded it because it doesn’t divide its efforts between a high-quality commercial version and a free ‘community’ version. The commercial and community teams collaborate to produce a single, high-quality release, which receives ongoing maintenance for a defined period. Both the release and ongoing updates are freely available to all users.

Ubuntu today has nine flavours and dozens of localised and specialised derivatives. There are also special editions for servers, OpenStack clouds, and mobile devices. All editions share common infrastructure and software, making Ubuntu a unique single platform that scales from consumer electronics to the desktop and up into the cloud for enterprise computing.

The Ubuntu OS and the innovative Ubuntu for Android convergence solution make it an exciting time for Ubuntu on mobile devices. In the cloud, Ubuntu is the reference operating system for the OpenStack project, it’s a hugely popular guest OS on Amazon’s EC2 and Rackspace’s Cloud, and it’s pre-installed on computers from Dell, HP, Asus, Lenovo and other global vendors. And thanks to that shared infrastructure, developers can work on the desktop, and smoothly deliver code to cloud servers running the stripped-down Ubuntu Server Edition.

After many years Ubuntu still is and always will be free to use, share and develop.

5.2.2 Network Simulator 2

NS is a discrete event simulator targeted at networking research. Ns provides substantial support for simulation of TCP, routing, and multicast protocols over wired and wireless (local and satellite) networks. The simulation tool used in the project is Network simulator 2 developed on the basis of refactoring in NS-1. In this version the use of OTcl (Object Tcl) was introduced.

The core of ns-2 is written in C++, but the C++ simulation objects are linked to shadow objects in OTcl and variables can be linked between both language realms. Simulation scripts are written in the OTcl language, an extension of the Tcl scripting language.

The general process of creating a simulation can be divided into several steps:

- Topology definition: to ease the creation of basic facilities and define their interrelationships.
- Model development: models are added to simulation (for example, UDP, IPv4, point-to-point devices and links, applications); most of the time this is done using helpers.
- Node and link configuration: models set their default ; most of the time this is done using the attribute system.
- Execution: simulation facilities generate events, data requested by the user is logged.
- Performance analysis: after the simulation is finished and data is available as a time-stamped event trace. This data can then be statistically analyzed with tools to draw conclusions.
- Graphical Visualization: raw or processed data collected in a simulation can be graphed using tools like XGRAPH.

The simulator I am using is installed on Virtual box (Vmware) wherein Ubuntu 12.04 LTS is given 10 GB memeory and NS 2.35 is installed on it.

5.3.3 Tcl and AWK script

Working on the project gave me the opportunity to work on Tcl and Awk script. In the simulation the Tcl script is used for network setup, parameters definition etc.

While the AWK script is used for measuring the network parameters like throughput, delay, jitters etc.

Tcl is a scripting language created by John Ousterhout. With programmers devising their own languages intended to be embedded into applications, Tcl gained acceptance on its own. It is commonly used for rapid prototyping, scripted applications, GUIs and testing. Tcl is used on embedded systems platforms, both in its full form and in several other small-footprint versions.

The combination of Tcl and the Tk GUI toolkit is referred to as Tcl/Tk.

AWK is an interpreted programming language designed for text processing and typically used as a data extraction and reporting tool. It is a standard feature of most Unix-like operating systems.AWK was created at Bell Labs in the 1970s.

The AWK language is a data-driven scripting language consisting of a set of actions to be taken against streams of textual data – either run directly on files or used as part of a pipeline – for purposes of extracting or transforming text, such as producing formatted reports.

The language extensively uses the string data type, associative arrays (that is, arrays indexed by key strings), and regular expressions.

While AWK has a limited intended application domain, and was especially designed to support one-liner programs, the language is Turing-complete, and even the early Bell Labs users of AWK often wrote well-structured large AWK programs.

CHAPTER 6. PROPOSED WORK

In case of wireless mesh networks the problem is that once an attack hampers the network the attack selectively targets packets of high importance which cause a sharp decline in the network performance and reduce the importance of the network.

Black Hole Attack

In networking, black holes refer to places in the network where incoming or outgoing traffic is silently discarded (or "dropped"), without informing the source that the data did not reach its intended recipient.

When examining the topology of the network, the black holes themselves are invisible, and can only be detected by monitoring the lost traffic.

In the network I aim to implement a wireless network where in first I will show the normal functioning of the network. Then calculate the network parameters like throughput, congestion window, packet delivery ratio, end to end delay etc.

Then I implement a black hole attack in the network which hampers the delivery of all the packets from the source to destination and instead send malicious packets.

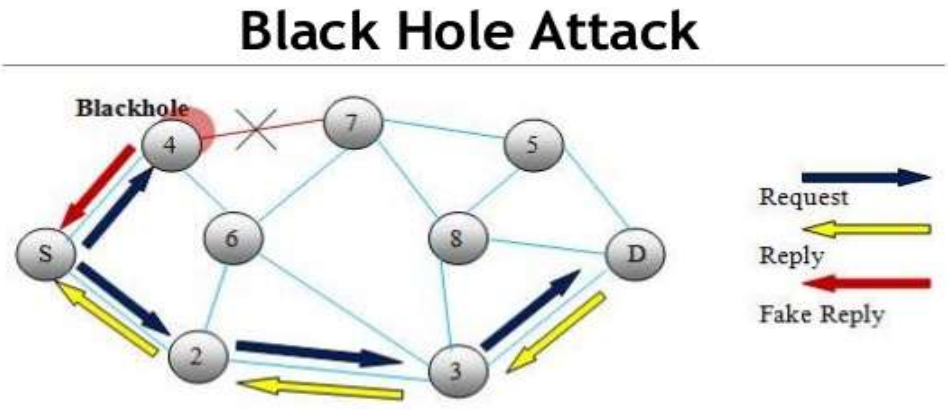


Fig Black Hole Attack

Now let us discuss what a black hole attack actually is, in computer networking we have packet drop attacks which can be either black hole or gray hole attacks.

Black hole attack is a type of denial-of-service attack in which a router that is supposed to relay packets instead discards them. This usually occurs from a router becoming compromised from a number of different causes.

The malicious router can also accomplish this attack selectively, e.g. by dropping packets for a particular network destination, at a certain time of the day, a packet every n packets or every t seconds, or a randomly selected portion of the packets.

This is rather called a gray hole attack. If the malicious router attempts to drop all packets that come in, the attack can actually be discovered fairly quickly through common networking tools such as traceroute

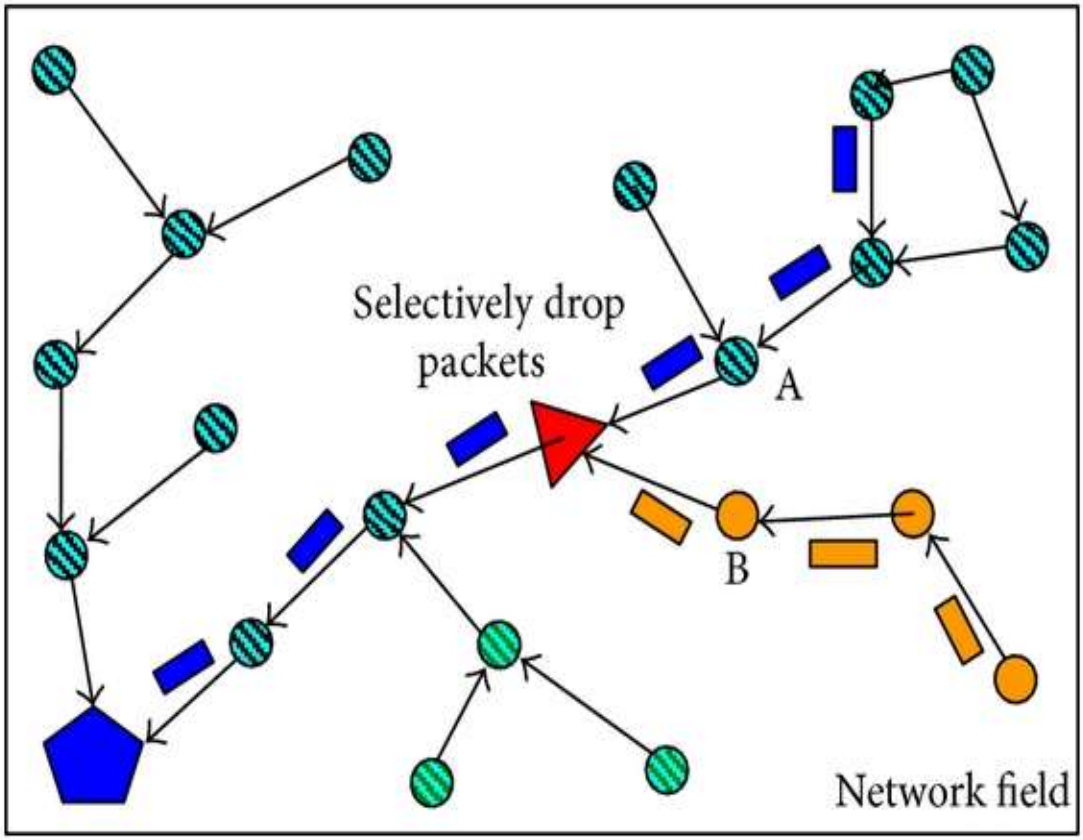


Fig. Gray Hole Attack

5.1 Pseudo-Code

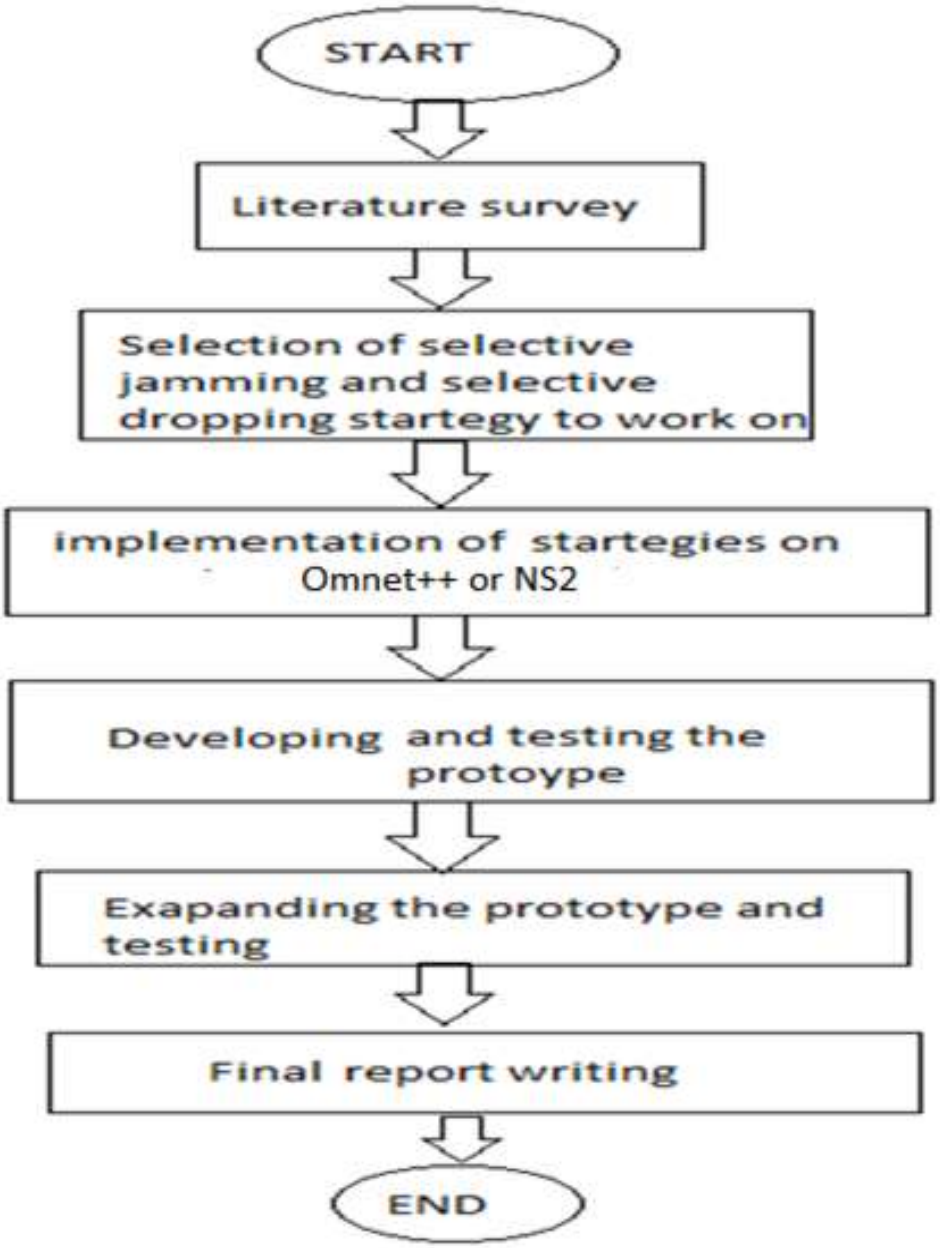
The algorithm implemented is :

- S is the node receiving a packet p for Destination D.
- N, is the set of one-hop neighbors.
- n, is a node of the set N that is used to forward the Packet.
- D is the destination of the packet.

```
if(Attack not implemented)
{
 use DSDV routing
 Forward Packet(p,n)
 return
}
else
{
 drop packet(p)
 forward malicious(X)
 return
}
```

Table Pseudo Code

5.2 Flow Chart



CHAPTER 7. SIMULATION RESULTS AND EVALUATION

Network simulator 2 (NS2) as been used to simulate and analyze the performance of the wireless networks. NS2 provides support for both wired, wireless and mobile networks but doesn't extend support for attack implementation in standard form .Therefore, the need to use an additional patch aroused, hence I used the Black hole patch that helps create an attack in the network.

The objective of this performance analysis is to find out how the performance parameters like throughput, packet delivery ratio and delay changes as our simulation scenario changes (i.e. the network expands).

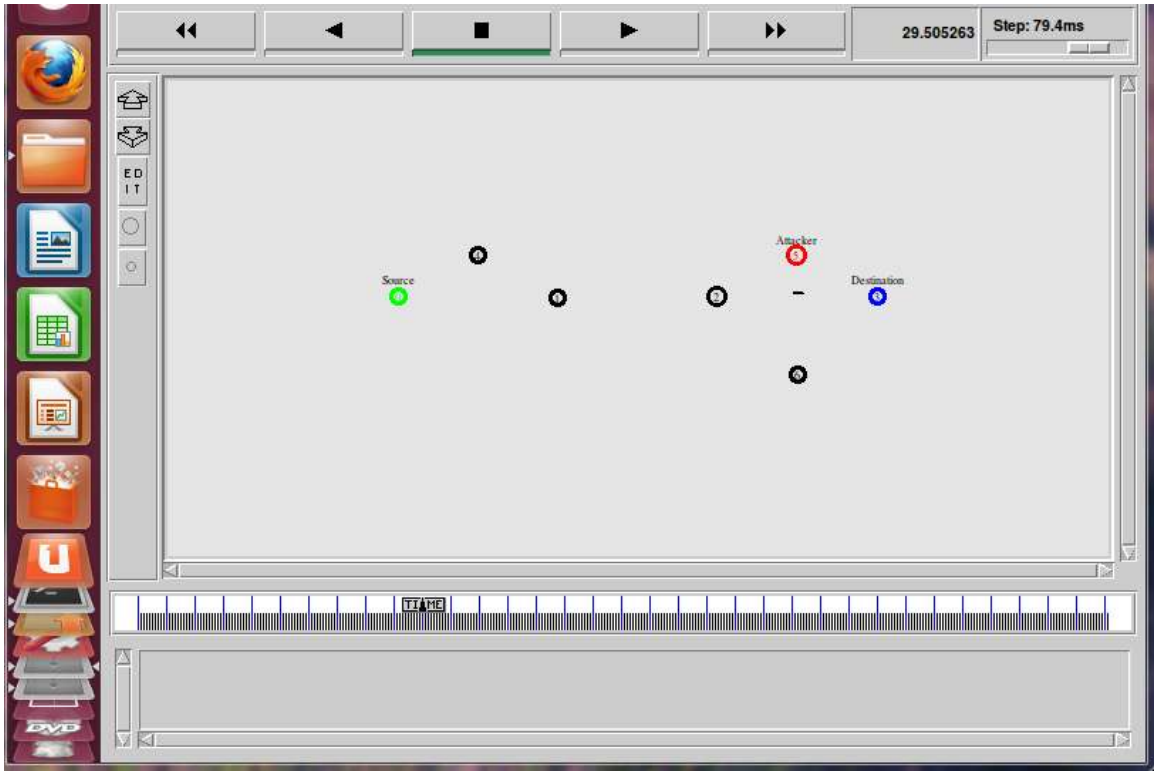
7.1 Simulation Environment

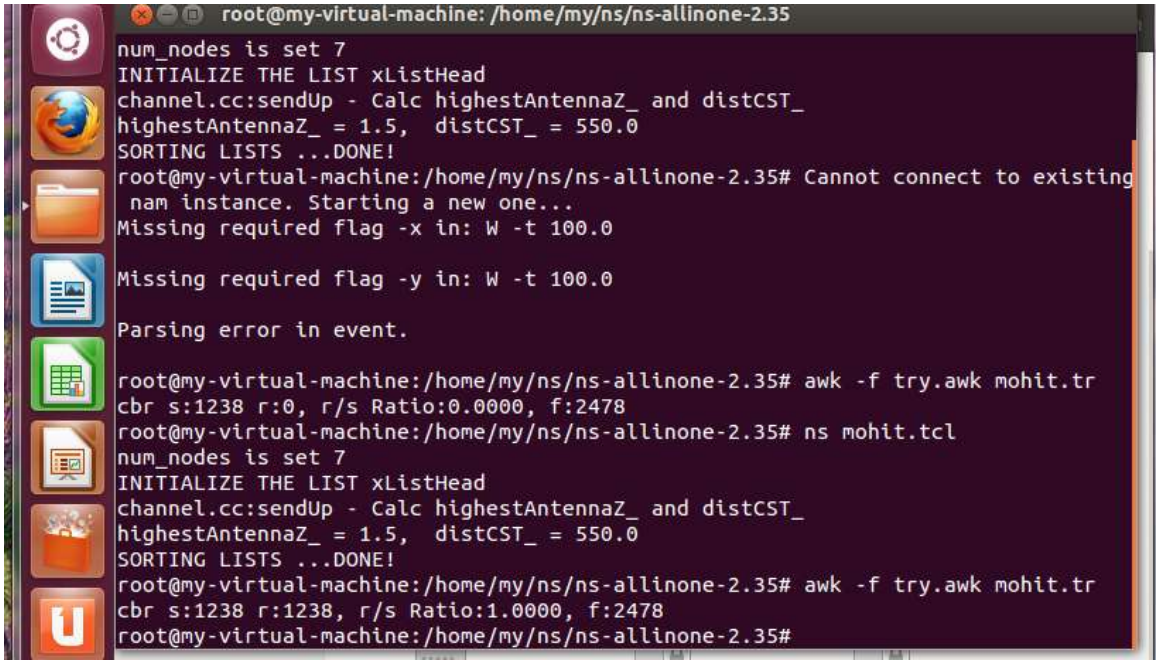
Simulation is for networks having 7, 10, 20 nodes in the network, of which some are mobile and some are stationary .we use identical simulation parameters for the three of them to get the true picture of the changes in throughput, delay and packet delivery ratio .

Nodes	Region
7	800 x 541
10	800 x 541
20	800 x 541

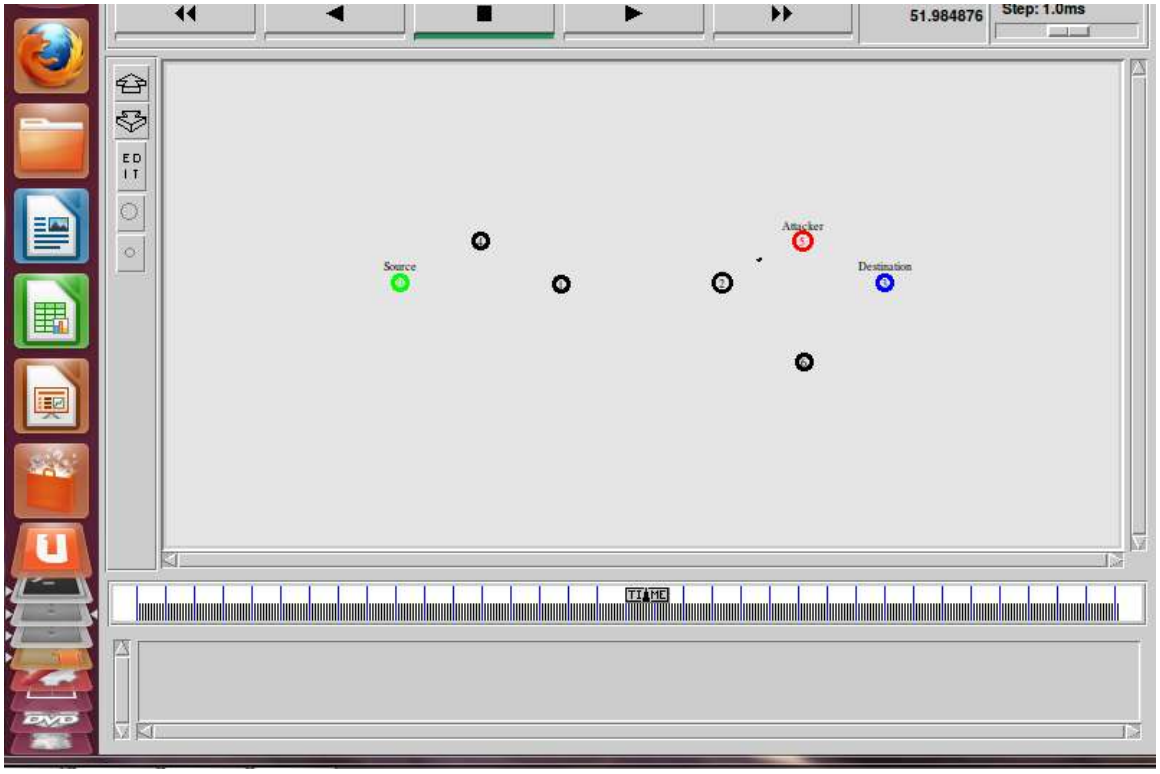
Table Node description

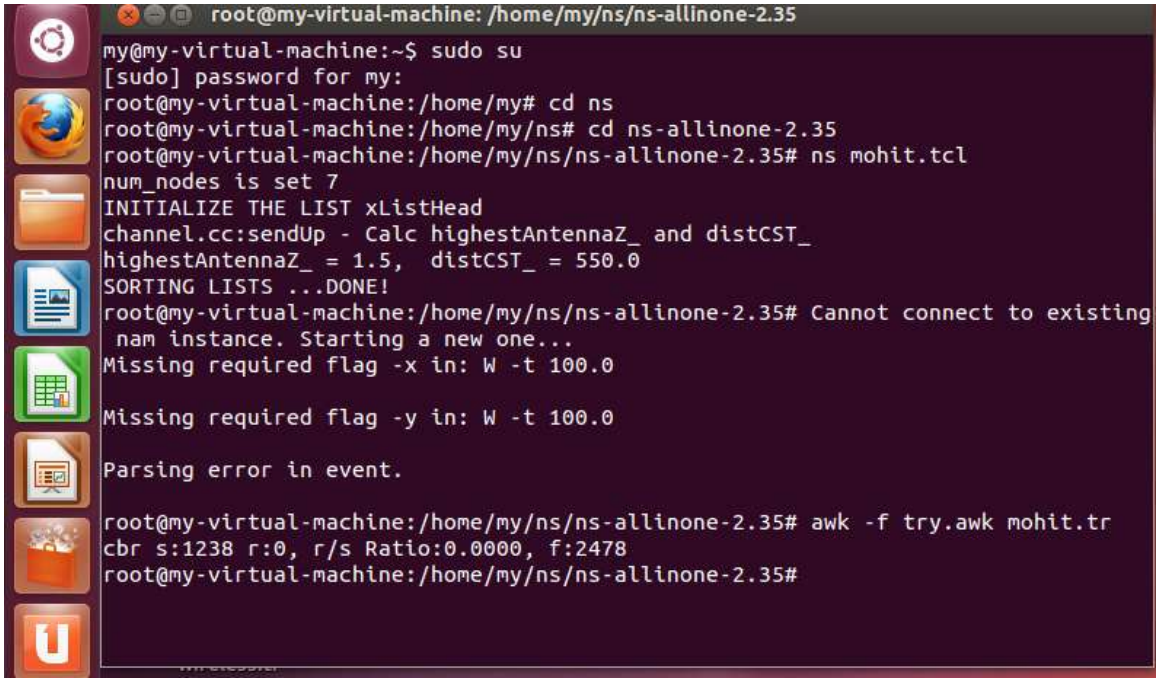
After the analysis of parameters like throughput, packet delivery ratio, end to end delay I found out that the effect of an attack occurring is more severe on larger networks and disrupts the functioning in a much worse way.





Normal functioning of network Attack not implemented.





Attacker attacking the network.

7.2 Packet Delivery Ratio

The ratio of the number of delivered data packet to the destination to the number of sent data packets. PDR illustrates the level of delivered data to the destination.

PDR = (∑ Number of packet receive / ∑ Number of packet send)

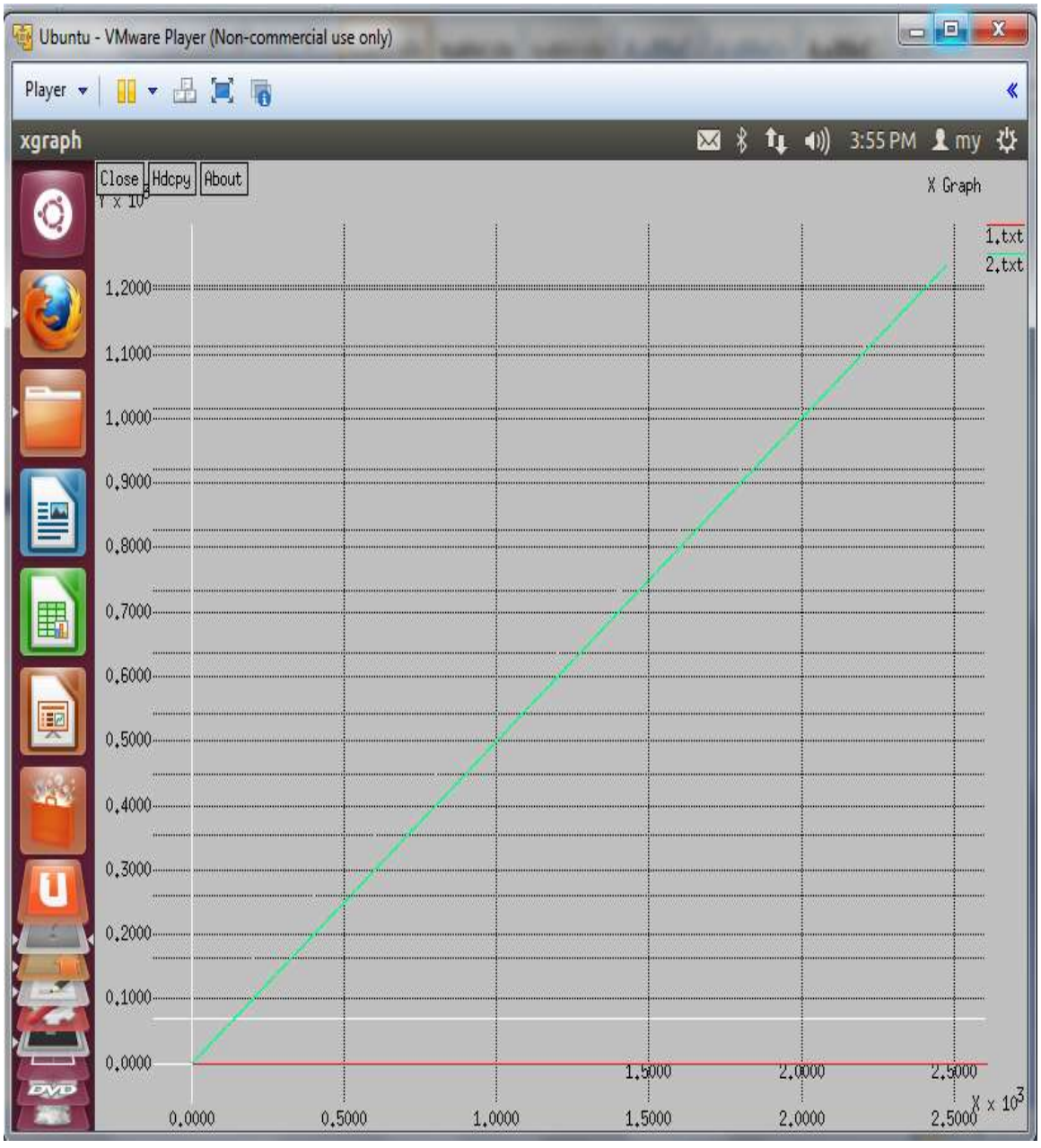


Fig Packet Delivery Ratio

7.3 Throughput

Throughput is the amount of data per unit time that is delivered from one node to another node via communication link. The throughput is measured in bits/second.

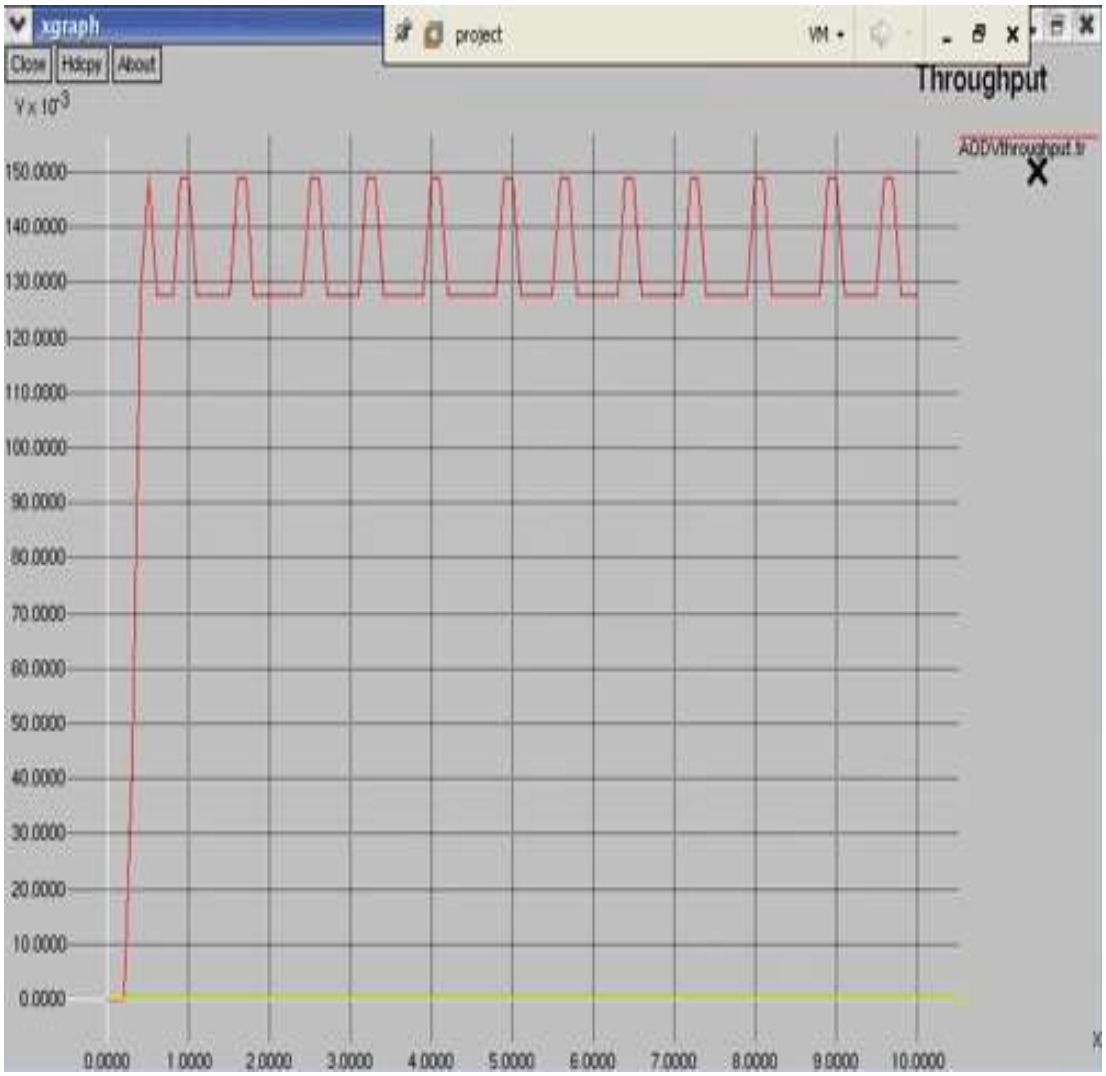


Fig Throughput

CONCLUSION

WMNs, although provide great connectivity and are the need today's age, are prone to different internal and external attacks. As discussed earlier that external attacks can be mitigated easily using a combination of cryptographic and robust communication techniques. It is the internal attacks that pose a problem as the network secrets are already available with the attacker. Jamming and sensing are two related functions in the physical-layer-based denial of services attacks.

The challenges in this project will be implementation of a random jamming node to stimulate the network. Development a mitigation strategy which can handle both selective jamming as well as selective dropping attacks is also an area for future work .

FUTURE WORK

The ubiquitous nature of wireless mesh networks increase their importance many folds. In this project we have so far seen as to how much can an attack hamper the network and compromise its network parameters. The future scope of this project includes development of strategy that mitigates the attack occurring and restores the normal functioning in the network.

The project can be extended further in terms of identifying a random jamming node that ruptures the network and eliminating that node and its effect out of the network hence restoring normal functioning.

I propose that this can be done using a different patch file that identifies the malicious node and cuts it out of the network.

REFERENCES

[1] A Proano, L Lazos , "Selective jamming attacks in wireless networks", IEEE International Conference on Communications(ICC) ,2010.

[2] Loukas Lazos and Marwan Krunz, "Selective jamming/Dropping Insider Attacks in wireless mesh networks", IEEE network, 2011.

[3] Mihail L. Sichitiu, "Wireless mesh networks: opportunities and challenges", World Wireless Congress, 2005.

[4] TX Brown, JE James, A Sethi , "Jamming and sensing of encrypted wireless ad hoc networks", 7th ACM international symposium on Mobile ad hoc networking and computing, 2006.

[5] K.Liu, J.Deng, P.K. Varshney and R.Balakrishnan, "An Acknowledgement–Based Approach for the Detection of Routing Misbehavior in MANETS", IEEE Transaction on Mobile Computing, 2011.

[6] JungFang Wang,Bin Xie, Dharma P. Agrwal, "Chapter 1 Journey from mobile Ad-hoc Networks to Wireless Mesh Networks", Guide to Wireless mesh Networks,Edition 1, Springer, pages 527.

[7] www.cs.tau.ac.il/~iftachh/Courses/FOC/Fall11/Slides/Commitments.pdf

Appendix

Code for AWK files:

Packet delivery ratio:

```
BEGIN {
	sendLine = 0;
	recvLine = 0;
	fowardLine = 0;
}

$0 ~/^s.* AGT/ {
	sendLine ++ ;
}

$0 ~/^r.* AGT/ {
	recvLine ++ ;
}

$0 ~/^f.* RTR/ {
	fowardLine ++ ;
}

END {
	printf "cbr s:%d r:%d, r/s Ratio:%.4f, f:%d \n", sendLine, recvLine,
(recvLine/sendLine),fowardLine;
}
```

Throughput:

```
BEGIN {
        recvdSize = 0
        startTime = 400
        stopTime = 0
    }
     {
            event = $1
            time = $2
            node_id = $3
            pkt_size = $8
            level = $4
      # Store start time
    if (level == "AGT" &;& event == "s" && pkt_size >= 512) {
    if (time <; startTime) {
            startTime = time
            }
        }
     # Update total received packets' size and store packets arrival time
    if (level == "AGT" &;& event == "r" && pkt_size >= 512) {
        if (time >; stopTime) {
```

```
        stopTime = time
            }
        # Rip off the header
        hdr_size = pkt_size % 512
        pkt_size -= hdr_size
        # Store received packet's size
        recvdSize += pkt_size
        }
    }
    END {
printf("AverageThroughput[kbps]=%.2f\t\tStartTime=%.2f\tStopTime=%.2f\n",(recvdSize/(stopTime startTime))*(8/1000),startTime,stopTime)
    }
```

Blackhole.patch

```
diff -Naur ns-allinone-2.35/ns-2.35/aodv/aodv.cc ns-allinone-2.35-modified/ns-
2.35/aodv/aodv.cc
--- ns-allinone-2.35/ns-2.35/aodv/aodv.cc      2010-04-30 22:40:36.000000000 +0530
+++ ns-allinone-2.35-modified/ns-2.35/aodv/aodv.cc          2014-05-23
22:16:05.000000000 +0530
@@ -82,6 +82,13 @@
       return TCL_OK;
     }

+// Added for Blackhole Attack[Code Starts]
+    if(strncasecmp(argv[1], "hacker", 6) == 0) {
+     malicious = true;
+      return TCL_OK;
+    }
+// Added for Blackhole Attack[Code Ends]
+
     if(strncasecmp(argv[1], "start", 2) == 0) {
       btimer.handle((Event*) 0);

@@ -144,7 +151,9 @@
   seqno = 2;
   bid = 1;

-  LIST_INIT(&nbhead);
+  malicious = false; // Added for Blackhole Attack
+
+LIST_INIT(&nbhead);
   LIST_INIT(&bihead);

   logtarget = 0;
@@ -439,6 +448,14 @@
 struct hdr_ip *ih = HDR_IP(p);
 aodv_rt_entry *rt;

+// Added for Blackhole Attack [Code Starts]
+// If the node is a malicious node - drop the packet and specify a reason for dropping it!
(Can't openly say you are malicious :-) )
+          if (malicious == true ) {
+             drop(p, DROP_RTR_ROUTE_LOOP); // DROP_RTR_ROUTE_LOOP is
added for no reason.
+             return;
+          }
+// Added for Blackhole Attack [Code Ends]
+
```

```
 /*
  *  Set the transmit failure callback.  That
  *  won't change.
@@ -758,6 +775,7 @@
   seqno = max(seqno, rq->rq_dst_seqno)+1;
   if (seqno%2) seqno++;

+
   sendReply(rq->rq_src,          // IP Destination
             1,                   // Hop Count
             index,               // Dest IP Address
@@ -767,8 +785,23 @@

   Packet::free(p);
  }
+
+// Added for Blackhole Attack [Code Starts]
+ else if (malicious == true) {
+    seqno = max(seqno, rq->rq_dst_seqno)+1;
+    if (seqno%2) seqno++;
+
+    sendReply(rq->rq_src,          // IP Destination
+              1,                   // Hop Count is set to 1 to confuse the source node!
+              rq->rq_dst,      // Dest IP Address
+              seqno,                       // Dest Sequence Num
+              MY_ROUTE_TIMEOUT,      // Lifetime
+              rq->rq_timestamp);     // timestamp
+   Packet::free(p);
+ }
+// Added for Blackhole Attack [Code Ends]

- // I am not the destination, but I may have a fresh enough route.
+// I am not the destination, but I may have a fresh enough route.

  else if (rt && (rt->rt_hops != INFINITY2) &&
                  (rt->rt_seqno >= rq->rq_dst_seqno) ) {
diff -Naur ns-allinone-2.35/ns-2.35/aodv/aodv.h ns-allinone-2.35-modified/ns-
2.35/aodv/aodv.h
--- ns-allinone-2.35/ns-2.35/aodv/aodv.h       2005-08-03 01:18:51.000000000 +0530
+++ ns-allinone-2.35-modified/ns-2.35/aodv/aodv.h 2014-05-23 21:02:55.000000000
+0530
@@ -273,6 +273,7 @@
           */
          double            PerHopTime(aodv_rt_entry *rt);
+         bool              malicious; // Added for Blackhole Attack
   nsaddr_t        index;   // IP Address of this node
```

Code for the network:

Blackhole.tcl :

```
set val(chan)   Channel/WirelessChannel    ;# channel type
set val(prop)   Propagation/TwoRayGround   ;# radio-propagation model
set val(netif)  Phy/WirelessPhy            ;# network interface type
set val(mac)    Mac/802_11                 ;# MAC type
set val(ifq)    Queue/DropTail/PriQueue    ;# interface queue type
set val(ll)     LL                         ;# link layer type
set val(ant)    Antenna/OmniAntenna        ;# antenna model
set val(ifqlen) 50                         ;# max packet in ifq
set val(nn)     7                          ;# number of mobilenodes
set val(rp)     AODV                       ;# routing protocol
set val(x)      800                        ;# X dimension of topography
set val(y)      541                        ;# Y dimension of topography
set val(stop)   100.0                      ;# time of simulation end

#===================================
#        Initialization
#===================================
#Create a ns simulator
set ns [new Simulator]

#Setup topography object
set topo       [new Topography]
$topo load_flatgrid $val(x) $val(y)
create-god $val(nn)

#Open the NS trace file
set tracefile [open blackhole.tr w]
$ns trace-all $tracefile

#Open the NAM trace file
set namfile [open blackhole.nam w]
$ns namtrace-all $namfile
$ns namtrace-all-wireless $namfile $val(x) $val(y)
set chan [new $val(chan)];#Create wireless channel

#===================================
#     Mobile node parameter setup
#===================================
$ns node-config -adhocRouting  $val(rp) \
                -llType        $val(ll) \
                -macType       $val(mac) \
```

```
            -ifqType       $val(ifq) \
            -ifqLen        $val(ifqlen) \
            -antType       $val(ant) \
            -propType      $val(prop) \
            -phyType       $val(netif) \
            -channel       $chan \
            -topoInstance  $topo \
            -agentTrace    ON \
            -routerTrace   ON \
            -macTrace      OFF \
            -movementTrace ON

#===================================
#       Nodes Definition
#===================================
#Create 7 nodes
set n0 [$ns node]
$n0 set X_ 99
$n0 set Y_ 299
$n0 set Z_ 0.0
$ns initial_node_pos $n0 20
set n1 [$ns node]
$n1 set X_ 299
$n1 set Y_ 297
$n1 set Z_ 0.0
$ns initial_node_pos $n1 20
set n2 [$ns node]
$n2 set X_ 499
$n2 set Y_ 298
$n2 set Z_ 0.0
$ns initial_node_pos $n2 20
set n3 [$ns node]
$n3 set X_ 700
$n3 set Y_ 299
$n3 set Z_ 0.0
$ns initial_node_pos $n3 20
set n4 [$ns node]
$n4 set X_ 199
$n4 set Y_ 350
$n4 set Z_ 0.0
$ns initial_node_pos $n4 20
set n5 [$ns node]
$n5 set X_ 599
$n5 set Y_ 350
$n5 set Z_ 0.0
```

```
$ns initial_node_pos $n5 20
set n6 [$ns node]
$n6 set X_ 600
$n6 set Y_ 200
$n6 set Z_ 0.0
$ns initial_node_pos $n6 20


# Node 5 is given RED Color and a label- indicating it is a Blackhole Attacker
$n5 color red
$ns at 0.0 "$n5 color red"
$ns at 0.0 "$n5 label Attacker"

# Node 0 is given GREEN Color and a label - acts as a Source Node
$n0 color green
$ns at 0.0 "$n0 color green"
$ns at 0.0 "$n0 label Source"

# Node 3 is given BLUE Color and a label- acts as a Destination Node
$n3 color blue
$ns at 0.0 "$n3 color blue"
$ns at 0.0 "$n3 label Destination"

#=====================================
#           Set node 5 as attacker
#=====================================
$ns at 0.0 "[$n5 set ragent_] hacker"

#=====================================
#       Agents Definition
#=====================================
#Setup a UDP connection
set udp0 [new Agent/UDP]
$ns attach-agent $n0 $udp0
set null1 [new Agent/Null]
$ns attach-agent $n3 $null1
$ns connect $udp0 $null1
$udp0 set packetSize_ 1500

#=====================================
#       Applications Definition
#=====================================
#Setup a CBR Application over UDP connection
set cbr0 [new Application/Traffic/CBR]
$cbr0 attach-agent $udp0
```

```
$cbr0 set packetSize_ 1000
$cbr0 set rate_ 0.1Mb
$cbr0 set random_ null
$ns at 1.0 "$cbr0 start"
$ns at 100.0 "$cbr0 stop"

#=====================================
#        Termination
#=====================================
#Define a 'finish' procedure
proc finish {} {
    global ns tracefile namfile
    $ns flush-trace
    close $tracefile
    close $namfile
    exec nam blackhole.nam &
    exit 0
}
for {set i 0} {$i < $val(nn) } { incr i } {
    $ns at $val(stop) "\$n$i reset"
}
$ns at $val(stop) "$ns nam-end-wireless $val(stop)"
$ns at $val(stop) "finish"
$ns at $val(stop) "puts \"done\" ; $ns halt"
$ns run
```