Islamic University–Gaza
Deanery of Higher Studies
Faculty of Engineering
Computer Engineering Department

Denial of Service Attack in Wireless Sensor Networks

Huda Bader Hubboub

Supervisor

Prof. Ibrahim S. I. Abuhaiba

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Science in Computer Engineering

1431H (2010)

ACKNOWLEDGMENT

All praise and glory are due to Allah the almighty who provided me with the much needed strength and stamina to successfully accomplish this work. The completion of this thesis also owes a great deal to the help and unrelenting support of the people around me. I am especially indebted to my advisor, Prof. Ibrahim Abuhaiba, for his guidance, support and patience with me throughout the course of my research. He taught me the essence and principles of research and guided me through until the completion of this thesis. I could not have asked for a better advisor.

Special thanks and appreciation are due to my friend Eng. Hanan Abu-Thuraya for all the helpful discussions all the way through.

I would like to express my gratitude to Mr. Mubashir Husain Rehmani from NS-2 mailing list for his useful suggestions during my work in NS-2 software.

I owe my largest debt to my family and I wish to express my heartfelt gratitude to all of them for their encouragement, constant prayers, and continued support. My dear parents who have given me all their love and support over the years; I thank them for their unwavering commitment through good times and hard times. My profound gratitude goes to my husband Jamal and my children Khaled and Demah for their support, endurance and patience. I dedicate this thesis to them.

Finally, I thank all of those whose names are not mentioned here but have helped me in any way to accomplish this work.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABELS

LIST OF ACRONYMS

WSN	Wireless Sensor Network
DoS	Denial of Service
DD	Directed Diffusion
FHSS	Frequency Hoping Spread Spectrum
SRAM	Static Random Access Memory
KB	Kilo Bytes
ACK	Acknowledgment
MAC	Media Access Control/Message Authentication Code
LAN	Local Area Network
RTS/CTS	Request-to-Send/Clear-to-Send
IEEE	Institute of Electrical and Electronics Engineers
Wi-Fi	Wireless Fidelity
ID	Identifier
BS	Base Station
TCP	Transmission Control Protocol
ICMP	Internet Control Message Protocol
AODV	Ad hoc On-Demand Distance Vector
TORA	Temporally-Ordered Routing Algorithm
EDD	Extended Directed Diffusion
SDD	Secure Directed Diffusion
LKHW	Logical Key Hierarchy for WSNs
LEAP	Localized Encryption and Authentication Protocol

µTESLA	Micro-Timed Efficient Stream Loss- tolerant Authentication
FLADS	Fuzzy Logic Anomaly Detection Scheme
ADV	Advertisement Message
MN	Master Node
GPS	Global Positioning System
JTAG	Joint Test Action Group
NS-2	Network Simulator-version 2
OTCL	Object Tool Command Language
CMU	Carnegie Mellon University
DCF	Distributed Coordination Function
CPU	Central Processing Unit
PDR	Packet Delivery Ratio

Denial of Service Attack in Wireless Sensor Networks

Huda Bader Hubboub

ABSTRACT

The objective of this thesis is to study the vulnerabilities of sensor networks, design, and implement new approaches for routing attack. As one of the cornerstones of network infrastructure, routing systems are facing more threats than ever; they are vulnerable by nature and challenging to protect.

In this thesis, we study different denial of service attack strategies against Directed Diffusion based WSNs. We introduce two new attacks. Reinforcement Swap Attack is our first attack which exploits the vulnerabilities of Directed Diffusion specifications. Its main idea is the disruption of configuration information, such as routing information to misuse route establishment along the network. Our approach is to swap Directed Diffusion reinforcement rule which means that the good route is excluded and the bad route is included. Moreover, our attack is activated and deactivated periodically to prolong its lifetime and hence brings down the target network. We present another attack, and call it Swarm Flooding Attack which targets the consumption of sensors computational resources, such as bandwidth, disk space, or processor time. Two variants of swarm attacks have been introduced namely Bee and Ant. Both approaches are inspired from the natural swarming difference between bees and ants. In all cases, the strategy used to mount an attack is the same. An attack consists of a set of malicious user queries represented by interests that are inserted into the network. However, the two forms of attack vary in the synchronization aspects among attackers. These types of attacks are hard to defend against as illustrated by past events (discussed within the thesis). For each of the proposed attack models, this thesis describes and presents analysis, simulation, and experimental measurements. We show that the system achieves maximal damage on system performance represented by sink throughput and average delay.

During this study, we analyze the parameter space of many possible denial of service attacks scenarios and make excessive simulations to identify what combination of parameter settings which leads to the more damaging and thus ultimate scenarios for our attack process.

Key Words: Wireless sensor network, denial of service attack, Directed Diffusion, on-off attack, swarming, flooding.

إنكار الخدمة في شبكات الاستشعار اللاسلكية

الملخص

الهدف من هذه الأطروحة هو دراسة نقاط الضعف لشبكات الاستشعار اللاسلكية وتصميم وتنفيذ طرق جديدة لإنكار الخدمة في شبكات الاستشعار اللاسلكية من خلال طبقة الشبكات. وباعتبارها واحدة من ركائز البنية التحتية للشبكة ، فإن بروتوكولات التوجيه تواجه المزيد من التهديدات أكثرمن أي وقت مضى فهي معرضة بطبيعتها للعديد من الأخطار.

في هذه الأطروحة، ندرس استراتيجيات مختلفة للهجوم ضد شبكات الاستشعار اللاسلكية التي تعتمد بروتوكول الانتشار الموجه للمعلومات أو ما يعرف ‘Directed Diffusion’ DD والمخصص لهذا النوع من الشبكات.

في هذه الدراسة، نقدم نوعين جديدين لإنكار الخدمة في شبكات الاستشعار اللاسلكية من خلال طبقة الشبكات المسئولة عن توجيه البيانات و التي تستخدم بروتوكول DD. النوع الأول، ويسمى استبدال قانون إنشاء طرق التوجيه، يعتمد على استغلال نقاط الضعف لبروتوكول DD الذي يطبق قانون التعزيز والمعاقبة لتحديد الطريق الأفضل لسير المعلومات. الفكرة الرئيسية لهذا الهجوم هي التأثير على معلومات إنشاء وتكوين طرق التوجيه التي سيتم من خلالها نقل البيانات لاحقاً بحيث يتم تعزيز الطريق السيء ومعاقبة الطريق الجيد وبالتالي يتم نقل البيانات من خلال الطرق البطيئة. أضف إلى ذلك، فإن هذا الهجوم يتم بشكل متقطع وليس مستمراً حيث يقوم المهاجِم بالتصرف وفق قانون التعزيز والمعاقبة الأصلي للبروتوكول لفترة من الزمن ثم يقوم بعكس القوانين في الفترة اللاحقة، وبالتالي يحدث التأثير المطلوب بالشبكة المستهدفة مع إمكانية بقاء المهاجِم فترة أطول دون أن يتم اكتشافه. أما النوع الاخر من الهجوم، والذي يتم من خلاله إغراق الشبكة بكمية كبيرة من الاستعلامات المزيفة (طلب للمعلومات) والتي يطلقها سرب من المهاجمين بحيث يتم تنسيق نوع هذه الاستعلامات ووقت إطلاقها. هذا الهجوم يستهدف بصورة أساسية استهلاك موارد شبكات الاستشعار، مثل عرض النطاق الترددي، مساحة الذاكرة، أو وقت المعالِج.

جميع هذه الأنواع من الهجمات من الصعب أن تكون شبكات الاستشعار اللاسلكية في مأمن منها كما تم إثباته من معظم الباحثين، وكما سيتم مناقشته في إطار هذه الأطروحة.

ومن خلال هذه الأطروحة، قمنا بوصف مفصل لكل نوع من النوعين المقترحين للهجوم، مع تقديم التحليل، والقيام بالتجارب لقياس أداء الهجوم وذلك من خلال برامج محاكاة الشبكات باستخدام الحاسب الالي. أظهرت النتائج أن كلاً من الهجومين المقترحين حقق الهدف المنشود بالحاق الضرر على أداء النظام، حيث تناقصت كمية المعلومات التي تصل للمستعِلم بشكل كبير بالإضافة إلى زيادة الوقت اللازم لكي تصل فيه هذه المعلومات. كذلك، قمنا بتحليل مفصل للعديد من السيناريوهات المحتملة للهجوم، وذلك لتحديد قيمة متغيرات النظام التي تحقق الضرر الأقصى وبالتالي الهجوم الأنجح من وجهة نظرنا.

الكلمات المفتاحية: شبكات الاستشعار اللاسلكية، هجوم إنكار خدمة الشبكات، بروتوكول الانتشار الموجه، الهجوم المتقطع.

Chapter 1
INTRODUCTION

1.1 Background

Wireless Sensor Network (WSN) is a wireless network of computing nodes that is formed automatically without human intervention. It potentially holds a very prominent place in technology history as the range of applications of these networks is astounding and includes military use, emergency response, surveillance, and scientific exploration of harmful environments; just to name a few. Each node in the network has the ability to discover its neighbors and to construct routes to reach other nodes in the collection. Like other networks, sensor networks are vulnerable to malicious attack; however, the hardware simplicity of these devices makes defense mechanisms designed for traditional networks infeasible. This work explores the Denial-of-Service (DoS) attack, in which a sensor node is targeted.

The aim of DoS attack is to make services unavailable to legitimate users, and current network architectures allow easy-to-launch and hard-to-stop DoS attacks. Particularly challenging are the service-level DoS attacks, whereby the victim links are destroyed and flooded with legitimate-like requests attack, in which wireless communication is blocked by malicious radio interference. These attacks are overwhelming even for massively-resourced services, and effective and efficient defenses are highly needed.

The main contribution of this work is the introduction of a new DoS attack framework against Directed Diffusion (DD) based WSN. In this thesis, we work exclusively with two different attack models, both of which are considered DoS attacks and we call them swap and swarm attacks. These attacks are used to show that we could affect the health of the network by utilizing the vulnerabilities of both wireless sensor network and the specifications of the DD protocol itself.

The first type of attack which we consider is defined as an On-Off Reinforcement Swap Attack and it focuses on swapping the rule of the control signaling used to establish the optimum route in DD protocol specifications. As different protocols specify different methods of setting up paths, it is fairly universal that when these operations are not performed properly or more precisely are completely and unkindly changed then a tremendous damage to the entire network may result. Moreover, our attack is not continuous, which means that malicious entities behave properly for a period of time in order to build up a strongly positive trust among other legitimate nodes, and then begin defecting for subsequent interval of time. This attack exploits the dynamic properties of trust through time-domain inconsistent behaviors in anti-attacks mechanisms.

We also introduce one more DoS attack namely Swarm Flooding Attack which integrates both concepts of flooding and swarming and involves sending large volumes of traffic to a victim system, to congest the victim system's network bandwidth with traffic. This causes the nodes that want to send application packet data to compete for the network's bandwidth, which in turn does not allow the network to communicate as normal as it should. By changing the attack parameters, new variants of the attack could be obtained such as Bee and Ant attacks which mainly differ in synchronization aspects between the attackers participated in the DoS process. Ant attack itself has more than one version. All of the proposed distinct attacking techniques result in significant degradation in system performance.

1.2 Wireless Sensor Network: An Overview

WSN is typically an ad hoc network of nodes with sensing abilities. So many routing protocols proposed for ad hoc networks could also be used for WSNs. The characteristics of WSNs are discussed from two perspectives: from the nodes that make up the network, and from the network itself.

In this chapter, we present basic information about wireless sensor networks; we start off by description of the characteristics of both sensor nodes and sensor network. Then, and as our research aims to attack this type of networks, we focus on presenting the most important vulnerabilities of sensor network.

1.2.1 Characteristics of Sensor Nodes

Sensor components: A WSN is typically composed of spatially distributed autonomous devices called nodes or motes which are scattered and together form a network that can perform tasks by communicating with each other using radio. A sensor network establishes a structure where it is possible to collect, process, analyze, and disseminate data [4]. In addition to one or more sensors, each sensor is equipped with a radio transceiver or other wireless communication device, a small microcontroller, and an energy source; usually a battery. A typical example is MICA2 sensors which have 128 KB flash instruction memory, 4 KB SRAM, an 8-bit microprocessor, and are powered by two AA batteries.

Different types of sensors for different applications*:* The development of wireless sensor networks was originally motivated by military applications such as battlefield surveillance. However, WSNs are becoming more commonplace and can be found in research projects and civilian applications as well as defense projects. The use of these WSNs can range from intrusion detection through production line monitoring to wildlife observation, and people are finding new and better ways to use the capabilities of WSNs every day. As a result, different types of sensors are available such as seismic, low sampling rate magnetic, thermal, visual, infrared, acoustic, and radar, which are able to monitor a wide variety of ambient conditions [5]. The nature of the sensor's application may affect the cost and physical size of the sensor nodes, but does not affect the general characteristics of WSNs (discussed in the next section).

From an operational point of view, it is also worth mentioning that sensor nodes might or might not have addressable global identification (ID). This fact affects how protocols and security schemes are designed for WSNs.

1.2.2 Characteristics of Wireless Sensor Networks

Large scale of deployment: a typical sensor network may consist of hundreds or thousands of heterogeneous nodes. In addition, sensor networks often have one or more points of centralized control called base stations. A base station is normally a gateway to another network, a powerful data processing or storage center, or an access point for human interface.

Resource scarcity*:* Particularly, a WSN is conceded to be formed of constrained-resources nodes which must be small; therefore, they are limited in power, memory and processing capacity. Energy is the most precious resource for sensor networks so communication is especially expensive in terms of power. For that reason, saving energy to prolong network life has a deep impact into the network architecture.

Multi-hop routing algorithm: A sensor network normally constitutes a wireless ad-hoc network, meaning that each sensor supports a multi-hop routing algorithm (several nodes may forward data packets to the base station).

Static/dynamic environments*:* where sensors are generally deployed in static, pre-determined locations with sensor readings taken at regular intervals and multi-hopped to a static sink for subsequent storage and analysis. Mobility extension in all its forms represents a more recent research subject in sensor networking, that is, mobility of sinks, mobility of sensors and actuators as well as mobility of code, i.e. applications.

Node failure recovery*:* As sensors usually are deployed in remote and hostile surroundings, people can not attend the sensor nodes. When some nodes fail due to exhausted batteries, faulted hardware and intrusion from attackers, these unattended nodes can not be changed or repaired. Failed nodes may lead to network partition which decreases the cover ratio reducing the availability of the network and even producing network failure. So, network topology should tolerate node-failure and activate self configuring schemes to avoid network partition [6].

1.2.3 Vulnerabilities of Wireless Sensor Networks

A wireless sensor network is a special network which has many constraints compared to a traditional computer network.

The large ad-hoc nature of wireless sensor networks adds some specific features and presents significant challenges in designing security schemes for wireless sensor networks.

Ad-hoc networks pose additional technical challenges in network discovery, network control and routing, collaborative information processing, querying, and tasking [7]. These ad hoc features combined with other factors make WSN vulnerable to different types of attacks. Below, we highlight some of these factors and features:

Wireless media is the most pronounced challenge where the pervasive applications proposed for WSNs necessitate wireless communication links. The wireless medium inherently increases the vulnerability of the network to eavesdropping, unauthorized access, spoofing, replay and denial-of-service (DoS) attacks. In addition, broadcast, which is an important communication primitive in wireless sensor networks, is highly desirable to broadcast commands (e.g., queries used to collect sensor data) and data (e.g., global clock value distributed for time synchronization) to the sensor nodes due to the large number of sensor nodes which adds more security challenges to sensory networks [8].

Conventional security solutions do not fit into sensor network system because these do not consider the structural specifics of resource restriction and recharge problem such as public-key cryptography [9]. These constraints limit the degree of encryption, decryption, and authentication that can be implemented on individual sensor nodes [10].

Physical capture risk which is added to the WSN as a result of being deployed in the field, so that sensor nodes are not tamper proof, i. e., they are susceptible to node capture attacks[11]. Attackers may capture a node, physically disassemble it, and extract from it valuable information (e.g. cryptographic keys). The highly hostile environment represents a serious challenge for security researchers. Although tamper proof hardware is available, it significantly increases cost and reduces the leeway for user/programmer error, as well as eliminates the reprogram ability [12]. Therefore, even with the existence of secure sensor network, it is highly assumed that nodes may be compromised by an attacker. Compromised nodes may exhibit arbitrary behavior and may collude with other compromised nodes. Similarly, an attacker can easily inject malicious messages into the wireless network.

Immense scale: Simply networking tens to hundreds of thousands of nodes has proven to be a substantial task. Providing security over such a network is equally challenging. Even with security aware protocols, security needs to scale to large-scale deployments. Most current standard security protocols were designed for two-party settings and do not scale to a large number of participants [13].

No central management/monitoring point: A sensor network should be a distributed network without a central management point. This will increase the vitality of the sensor network. However, perhaps most importantly, the longer that a sensor is left unattended, the more likely that an adversary has compromised the node [14]. Also, this makes it difficult to detect the attack and even if the mote discovers that, it is being attacked, it is not easy for it to survive due to its limited capabilities.

Deployment mechanism of sensors: where the random aerial deployment is the most common procedure in the majority of sensory applications. And although the coverage area of interest is supposed to be 100% covered by the densely deployed nodes, random deployment may have coverage holes; areas not covered by any node, due to random aerial deployment creating voids, presence of obstructions, and, more likely, node failures [15].

All these limitations make sensor networks more vulnerable to attacks, ranging from passive eavesdropping to active interference. In particular, we distinguish attacks as outsider and insider attacks. In outsider attacks, the attacker may inject useless packets in the network in order to exhaust the energy levels of the nodes, or passively eavesdrop on the network's traffic and retrieve secret information. An insider attacker, however, compromises a legitimate sensor node and uses the stolen key material, code, and data in order to communicate with the rest of the nodes, as if it was an authorized node. With this kind of intrusion, an attacker can launch more powerful and hard to detect attacks that can disrupt or paralyze the network [16].

1.3 Research Overview

In this section, we present detailed information about this thesis. First, we start by identifying the importance of this field in sensor networking including the motivations behind our study, objectives to be accomplished, methodology that has been followed, our contributions throughout this work and finally, we show the content of this research.

1.3.1 Motivation

A typical wireless sensor network is expected to give a certain data that the user is actively enquiring about after some amount of time. Many attack schemes tend to stop the proper performance of sensor networks to delay or even prevent the delivery of data requested by user. Despite the fact that the term attack usually refers to an adversary's attempt to disrupt, undermine, or destroy a network, a Denial-of-Service (DoS) attack refers to any event that diminishes or eliminates a network's ability to perform its expected function [1]. Such a technique may be helpful in specific applications such as utilizing the best of these attacks to find the weak tips of presented protocols at different layers. These attacks consequently would expose weaknesses that lead to effective countermeasures. Understanding these vulnerabilities can develop techniques for identifying attacks that attempt to take advantage of them and implement mechanisms to mitigate these attacks. In other more serious application, there are situations where network blocking is necessary to protect public safety. For example, in hostile environments disabling the communication capabilities of the enemy represents a high priority. Another example is to prevent cell phone detonation of bombs. Furthermore, denial of service attack can be used in legitimate scenarios to achieve such purpose at different layers of the protocol. However, we chose to exploit the routing layer which represents one of the famous techniques widely used for this.

1.3.2 Objectives

This research explores the capabilities, strengths, and weaknesses of injecting multiple spreading attackers in WSN and the associated performance.

The fundamental goal for this research is twofold. First, it explores a class of vulnerabilities in routing layer of wireless sensor networks in which sensor networks can be rapidly clogged by a malicious DoS attack. Second, we focus our study to select one of these routing protocols as a model to identify and analyze its performance on the presence of DoS attack. We select one famous routing protocol in WSN namely Directed Diffusion which is data centric dissemination protocol for sensor networks [2].

To achieve these goals of designing a strong DoS attack to degrade the performance of routing protocol, NS-2, network simulator [3], is used as a simulation tool to evaluate the performance of both on-off reinforcement swap and swarm flooding attacks. This simulation provides valuable insight into system performance under various operating conditions. The relative performance improvements and under what circumstances these occur can then be studied during this research. The results can be used to optimize the attacks under various conditions.

1.3.3 Methodology

This research was conducted in the following phases:

- Study the basic principal of wireless sensor networks.
- Study the principle of routing in WSNs, mainly Directed Diffusion DD protocol.
- Review of the existing Denial of Service attack schemes.
- Review the existing related work of DD.
- Identify the major security threats and attacks of DD.
- Find out the major points of DD vulnerability.
- Design the proposed DoS attacks.
- Evaluate the design using NS-2, network simulator.

1.3.4 Contributions

This thesis explores the effect of denial of service attacks on wireless sensor network. With time, it is expected that these attacks would become more sophisticated, potent and increase their impact on degrading the network performance. Hence, it is all the more necessary for developers to incorporate methods of preventing these attacks during design time rather than patching up the system once the attack has been made. The contributions of this research are highlighted hereunder:

- One of the goals of this thesis is to raise awareness of the impact of denial of service attacks on sensor networks so that a defense mechanism can be put in place much before such attacks become widespread.

- We present a detailed description of several DoS attacks performed by possibly colluding adversaries. We also review the techniques used to mitigate these attacks.
- We introduce two new attacks against Directed Diffusion based WSN; the first attack utilizes the vulnerability of the routing protocol itself, while the second attack is general and could be applied to any other routing protocol.
- We investigate the impact of different forms of these attacks which were implemented on NS-2 simulator. We demonstrate through simulation the effects of the presented attacks on the Directed Diffusion routing protocol. Our results quantify the damage caused by the attacks and provide insights into identifying those which result in the greatest network disruption while requiring the least number of adversarial participants.
- We provide what we believe to be the first formula to estimate the value of the number of attackers for a given number of legitimate nodes in the network using the connectivity rules. Based on our knowledge, no one has previously used any formula to figure out the appropriate number of attackers.

1.3.5 Organization

We commence this thesis by providing a background in the area of wireless networking mainly the characteristics and vulnerabilities of WSNs. Also, this chapter contains a brief review about this research. This includes the denial-of-service problem, along with research objectives, methodology, contributions and provides an overview of the remainder of this document. In Chapter 2, we address the classes of denial-of-service attack at different layers of the protocols. This part also intends to provide a brief survey of known, and published common defense and mitigation strategies used in this field. Chapter 3 is devoted to describe Directed Diffusion, a technique for providing low-energy consumption routing in sensor networks on which our design and evaluation is based. For comparison reasons, we also include the all related work to DD concerning security and attacks. Afterwards in Chapter 4, which contains comprehensive and formal description of our work, we propose two algorithms for performing DoS attack (reinforcement swap, and swarm flooding). It goes on to discuss the whole attack process including the introduction of a new formula to estimate the approximate number of attackers needed to launch that attack effectively. It covers all relevant details about the background of these attacks. Chapter 5 explains the evaluation of this research effort via simulation. This includes the discussion and choice of relevant parameters and all the specifications of the implemented attacks. Chapter 6 concludes the thesis with a summary of the research findings, including important concepts, techniques behind this research effort, and the significance of research results. Finally, the chapter presents recommendations for future direction of the research area.

Chapter 2

DENIAL OF SERVICE ATTACK IN WIRELESS SENSOR NETWORKS

2.1 Denial of Service Attack: An Overview

The very idea of a wireless network introduces multiple venues for attack and penetration that are either much more difficult or completely impossible to execute with a standard, wired network. This inherent limitation makes WSNs especially sensitive to several key types of attacks. In contrast to resource-rich networks such as the Internet, a WSN is less stable, more resource-limited, subject to open wireless communication, and prone to the physical risks of in-situ deployment. These factors increase the susceptibility of WSNs to distinct types of attacks.

Although there are many factors (software and hardware bugs, environmental conditions) that could diminish the capacity of the network to provide the requisite service, there is the possibility that the service is denied as a result of being attacked by an adversary.

Attacks can be performed in a variety of ways, most notably as denial of service attacks, but also through traffic analysis, privacy violation, node takeover, attacks on the routing protocols, and attacks on a node's physical security which all are out of the scope of this research. In this chapter, we first address some common denial of service attacks and then describe the most famous defensive strategies against them.

A Denial of Service (DoS) attack can be defined in many forms. In [17], the authors describe DoS attack as an incident in which a user is deprived of the services of a resource he would normally expect to have. While in [1] it was defined as "any event that diminishes or eliminates a network's capacity to perform its expected function". In general, DoS is meant not only for the adversary's attempt to subvert, disrupt, or destroy a network, but also for any event that diminishes a network's capability to provide a service. In wireless sensor networks, several types of DoS attacks in different layers might be performed.

Different types of DoS attacks in different layers of a sensor network protocol stack are discussed in Wood and Stankovic [1], and some countermeasures to defend against them are proposed. Security problems of different sensor network routing protocols are analyzed and mechanisms to enhance the security of sensor network routing are proposed in Karlof and Wagner [18].

2.2 Denial of Service Attacks in Unsecure WSNs

Sensor networks are usually divided into layers, and this layered architecture makes WSNs vulnerable to DoS attacks as DoS attacks may occur in any layer of a sensor network. Stankovic and Wood particularly look into protocol level vulnerabilities [1]. This section enlists the layered network architecture of the wireless sensor networks, the denial of service vulnerabilities of each layer and the most famous defenses possible against these attacks.

Physical Layer:

As with any radio based medium, there exists the possibility of jamming in WSNs. In addition, nodes in WSNs may be deployed in hostile or insecure environments where an attacker has an easy physical access. These two vulnerabilities are explored as follows:

1. <u>Jamming</u>: Jamming is a type of attack which interferes with the radio frequencies that network's nodes are using [19, 20]. A jamming source may either be powerful enough to disrupt the entire network or less powerful and only able to disrupt a smaller portion of the network. Even with lesser powered jamming sources, such as a small compromised subset of the network's sensor nodes, an adversary has the potential to disrupt the entire network provided the jamming sources are randomly distributed in the network.

Typical defenses against jamming involve variations on spread-spectrum communication such as frequency hopping and code spreading. Frequency-hopping spread spectrum

(FHSS) [21] is a method of transmitting signals by rapidly switching a carrier among many frequency channels using a pseudo random sequence known to both transmitter and receiver. Without being able to follow the frequency selection sequence an attacker is unable to jam the frequency being used at a given moment in time. However, as the range of possible frequencies is limited, an attacker may instead jam a wide section of the frequency band.

Code spreading is another technique used to defend against jamming attacks and is common in mobile networks. However, this technique requires greater design complexity and energy restricting its use in WSNs. In general, to maintain low cost and low power requirements, sensor devices are limited to single-frequency use and are therefore highly susceptible to jamming attacks.

2. Tampering: Another physical layer attack is tampering [1, 22]. Given physical access to a node, an attacker can extract sensitive information such as cryptographic keys or other data on the node. The node may also be altered or replaced such that a malicious code replaces the legitimate one to create a compromised node which the attacker controls. One defense to this attack involves tamper-proofing the node's physical package. Indeed, there are many advantages of tamper-proofing other than obstructing attacks. For example, a tamper-proof body could prevent wear and tear due to environmental factors, thereby increasing the overall life of the node assuming we have sufficient power availability. The main hindering factor of building sensor nodes in tamper-proof bodies is cost. Hence, it is usually assumed that the sensor nodes are not tamper-proofed in WSNs due to the needed additional cost.

Link Layer:

1. Collisions: A collision occurs when two nodes attempt to transmit on the same frequency simultaneously [1]. When packets collide, a change will likely occur in the data portion causing a checksum mismatch at the receiving end. The packet will then be discarded as invalid. An adversary may strategically cause collisions in specific packets e.g., IEEE 801.11b (Wi-Fi) protocol by continually transmitting messages in an attempt to

generate collisions. Such collisions would require the retransmission of any packet affected by the collision such as ACK control messages. A possible result of such collisions is the costly exponential back-off in certain MAC (Media Access Control) protocols. Using this technique it would be possible for an attacker to simply deplete a sensor node's power supply by forcing too many retransmissions.

A usual defense against collisions is the use of error correcting codes [1]. However, most codes work best with low levels of collisions such as those caused by environmental or probabilistic errors. As these codes add processing and communication overhead, their complexity and therefore effectiveness has a low upper bound. It is reasonable to assume that an attacker will always be able to corrupt more than what can be corrected. While it is possible to detect these malicious collisions, no complete defenses against them are known at this time.

2. Exhaustion: of network resources by inducing repeated retransmission attempts. Even in the absence of high-rate traffic, if a node must continually retransmit due to collisions, as it is naively implemented in link layer, or has to route heavy traffic, eventually its energy may be exhausted [23].

A possible solution is to apply rate limits to the MAC admission control such that the network can ignore excessive requests preventing the energy drain caused by repeated transmissions. A second technique is to use time-division multiplexing where each node is allocated in a time slot in which it can transmit. This eliminates the need of arbitration for each frame and can solve the indefinite postponement problem in a back-off algorithm. However, it is still susceptible to collisions.

3. Unfairness: For data-collection applications in sensor networks, it is important to ensure all data sources have weighted access to network bandwidth so that the base stations receive a complete picture about the monitored area [24]. On the contrary, unfairness which is considered a weak form of a DoS attack [1] can be performed by attacker attempt to degrade the network performance instead of completely preventing access to a service. An attacker targets fairness by irregular experience of some of the abovementioned link

layer attacks. An attacker can only degrade it to give them an advantage such as causing other nodes in a real-time MAC protocol to miss their transmission deadline. The use of small frames lessens the effect of such attacks by reducing the amount of time an attacker can capture the communication channel. However, this technique often reduces efficiency and is susceptible to further unfairness such as an attacker trying to retransmit quickly instead of randomly delaying.

4. Interrogation: This makes use of the interaction that takes place between two nodes prior to data transmission. For example, wireless LANs (IEEE 802.11) use Request to Send (RTS) and Clear to Send (CTS). An attacker can exhaust a node's resources by repeatedly sending RTS messages to elicit CTS responses from a targeted neighbor node [1]. To put a defense against such type of attacks a node can limit itself in accepting connections from same identity or use anti replay protection and strong link-layer authentication [25].

Network Layer:

1. Spoofed, altered or replayed routing information: The straightest attack against a routing protocol in any network is to target the routing information itself as it is exchanged between nodes. An attacker may spoof, alter, or replay routing information in order to disrupt traffic in the network [14]. These disruptions include the creation of routing loops, attracting network traffic from select nodes, extending and shortening source routes, generating fake error messages, partitioning the network, and increasing end-to-end latency.

A countermeasure against spoofing and alternation is to append a MAC (Message Authentication Code) after the message. By adding a MAC to the message, the receivers can verify whether the messages have been spoofed or altered. To defend against replayed information, counters or timestamps can be included in the messages.

2. Selective forwarding: A basic guess made in multi-hop networks is that all nodes in the network will truthfully forward receive messages. A specific form of this attack is the black hole attack in which a node drops all messages it receives as if the node doesn't exist

at all. An attacker may perform another form of attack by selectively forwarding only certain messages and simply dropping others which is denoted by grey holes [18, 26]. One defense against selective forwarding attacks is using multiple paths to send data. A second defense is to detect the malicious node or assume it has failed and seek an alternative route. A third alternative defense is to use implicit acknowledgments, which ensure that packets are forwarded as they were sent [18, 25].

3. Sinkhole: In a sinkhole attack, an attacker makes a compromised node look more attractive to nearby nodes by forging routing information [1, 18, 25]. The end effect is that surrounding nodes will choose the compromised node as the next node to route their data through. This type of attack makes selective forwarding very simple as all traffic from a large area in the network will flow through the adversary's node.

Geo-routing protocols are identified as one of the routing protocol groups that are resistant to sinkhole attacks, because the topology is built using only localized information, and traffic is naturally routed based on the physical location of the sink node, which makes it difficult to lure it elsewhere to create a sinkhole [18, 25, 27].

4. Sybil: In this attack, a single node presents a variety of identities to all other nodes in the WSN. This may mislead other nodes, and hence routes believed to be disjoint with respect to node can have the same adversary node. Protocols and algorithms which are easily affected include fault-tolerant schemes, distributed storage, and network topology maintenance. For example, a distributed storage scheme may rely on there being three replicas of the same data to achieve a given level of redundancy. If a compromised node pretends to be two of the three nodes, the algorithms used may conclude that redundancy has been achieved while in reality it has not. A countermeasure to Sybil Attack is by using a unique shared symmetric key for each node with the base station [25, 27].

5. Wormholes: An adversary can tunnel messages received in one part of the network over a low latency link and replay them in another part of the network. This is usually done with the coordination of two adversary nodes, where the nodes try to understate their distance from each other, by broadcasting packets along an out-of-bound channel available only to

the attacker. This link may either be a single node forwarding messages between two adjacent but otherwise non-neighboring nodes or a pair of nodes in different parts of the network with the ability to communicate between each other. The latter of these cases is closely related to the sinkhole attack as an attacking node near the base station can provide a one hop link to that base station via the other attacking node in a distant part of the network [28].

To overcome this, the traffic is routed to the base station along a path, which is always geographically shortest or use very tight time synchronization among the nodes, which is infeasible in practical environments [27].

6. Hello flood attacks: This attack exploits Hello packets that are used in many protocols to declare nodes to their neighbors. A node receiving such packets may believe that it is in radio range of the sender. A laptop attacker may use a high powered transmitter to trick a large area of nodes into believing they are neighbors of that transmitting node [18]. If the attacker falsely broadcasts a superior route to the base station, all of these nodes will attempt transmitting to the attacking node despite many being out of their radio range in reality.

Authentication is the key solution to such attacks. Such attacks can easily be avoided by verifying bi-directionality of a link before taking action based on the information received over that link [27, 28]

7. Acknowledgement spoofing: Routing algorithms used in sensor networks sometimes require acknowledgements to be used. An attacking node can spoof the acknowledgements of overheard packets destined for neighboring nodes in order to afford false information to those neighboring nodes [18]. A case of such false information is claiming a node is alive when in fact it is dead. The most apparent solution to this problem would be authentication via encryption of all sent packets and also packet headers [25, 27].

8. Misdirection: This is a more active attack in which a malicious node present in the routing path can send the packets in wrong direction through which the destination is

unreachable. Instead of sending the packets in the correct direction, the attacker misdirects these packets towards one victim node. If it gets observed that a node's network link is getting flooded without any useful information then the victim node can be scheduled into sleep mode for some time to overcome this [27].

Transport Layer:

Two possible attacks in this layer, flooding and de-synchronization, are discussed in this subsection.

1) Flooding: Whenever a protocol is required to maintain state at either end of a connection it becomes vulnerable to memory exhaustion through flooding [26]. An attacker may repeatedly make new connection requests until the resources required by each connection are exhausted or reach a maximum limit. A very common form of DoS attacks involves sending a large number of common packets aimed at a single destination. The most common packets used are: TCP, ICMP, and UDP. The huge traffic deluge caused by these packets leads the network to no longer be able to distinguish between legitimate and malicious traffic. Basically all available resources such as bandwidth are used up and nothing is left for legitimate use causing the users to be denied the service of the network.

One proposed solution to this problem is to require that each connecting client demonstrate its commitment to the connection via solving of a puzzle. The idea is that a connecting client will not needlessly waste its resources creating unnecessary connections. Given an attacker does not likely have infinite resources, it will be impossible for them to create new connections fast enough to cause resource starvation on the serving node. While these puzzles do include processing overhead, this technique is still more desirable than excessive communication.

2) De-synchronization: De-synchronization refers to the disruption of an existing connection [18]. An attacker may, for example, repeatedly spoof messages to an end host causing that host to request the retransmission of missed frames. And if the adversary maintains a proper timing, an attacker may degrade or even prevent the ability of the end

hosts to successfully exchange data to instead waste energy attempting to recover from errors which never really existed. This will cause a considerable drainage of energy of legitimate nodes in the network in an endless synchronization-recovery protocol. A possible solution to this type of attack is to require authentication of all packets including control fields communicated between hosts. Header or full packet authentication can defeat such an attack [27, 28].

In most cases these attacks are made more effective by changing a few attributes on the packets. For example, false source IP addresses could be embedded into the packets (IP Spoofing) so that the server can't find the sender when it is trying to communicate. Type of Denial of Service attack can be envisioned. If these devices are made to execute an invalid energy hungry program repeatedly, it will cause their battery to drain out much before their expected life time [28, 29].

Application Layer:

1) Overwhelm attack: An attacker might attempt to overwhelm network nodes with sensor stimuli, causing the network to forward large volumes of traffic to a base station. This attack consumes network bandwidth and drains node energy. This attack can be mitigated by carefully tuning sensors so that only the specifically desired stimulus, such as vehicular movement, as opposed to any movement, triggers them. Rate-limiting and efficient data-aggregation algorithms can also reduce these attacks' effects [27].

2) Path-based DOS attack: It involves injecting spurious or replayed packets into the network at leaf nodes. This attack can starve the network of legitimate traffic, because it consumes resources on the path to the base station, thus preventing other nodes from sending data to the base station. Combining packet authentication and anti replay protection prevents these attacks [18].

3) Deluge (reprogram) attack: Network-programming system lets you remotely reprogram nodes in deployed networks. If the reprogramming process isn't secure, an intruder can

hijack this process and take control of large portions of a network. It can use authentication streams to secure the reprogramming process [63].

2.3 Denial of Service Attack against Authenticated WSNs

As previously explored, authentication was mentioned as a primary solution to multiple attacks in different layers as it prevents unauthorized access to the network. Authentication can be provided using cryptographic algorithms. But the protocols incorporating these algorithms must be efficient enough to not become the target of battery exhaustion attacks themselves. There are two general approaches for broadcast authentication in wireless sensor networks: digital signatures and µTESLA-based techniques. However, both signature-based and µTESLA-based broadcast authentications are themselves vulnerable to Denial of Services (DoS) attacks as viewed next:

DoS Attacks against Signature-Based Broadcast Authentication WSNs

Although it is possible to perform digital signature operations on sensor nodes, the cost of such operations is still substantially higher than that of symmetric cryptographic operations, and will substantially consume the battery power if frequently performed. This leads to a fatal threat to signature-based broadcast authentication: An attacker may simply forge a large number of broadcast messages with digital signatures, force sensor nodes to receive these packets, verify their signatures, and eventually deplete their battery power. Benign sensor nodes may certainly decide not to forward broadcast messages before their signatures are verified. However, a single malicious node can still overload and disable many benign nodes in its local region with forged messages. Moreover, an attacker may generate much higher impact by increasing the signal strength or deploying multiple malicious nodes in different locations [8].

DoS Attacks against µTESLA-Based Broadcast Authentication

A major limitation of µTESLA and its variations is the authentication delay. In other words, a receiver can not authenticate a broadcast packet immediately after receiving it. Note that a broadcast packet typically has to be forwarded (via local re-broadcast) multiple times before it reaches all the nodes. This means that a sensor node has to forward a broadcast packet before properly authenticating it. The key disclosed in a broadcast packet can provide some weak authentication. However, once an attacker receives a broadcast packet, he/she can reuse this key to forge many packets that can pass this weak authentication. As a result, similar to the DoS attacks against signature-based broadcast authentication, an attacker can force regular nodes to forward a large number of bogus packets to eventually exhaust their battery power [8].

2.4 Defense Strategies against DoS Attack in WSNs

The mechanisms to prevent DoS attacks include payment for network resources, pushback, strong authentication and identification of traffic [14]. Currently, there are four basic mechanisms that could be helpful to overcome DoS attacks in sensor networks.

Watchdog scheme: A necessary operation to overcome DoS attacks is to identify and circumvent the misbehaving nodes [30]. Watchdog scheme attempts to achieve this purpose through using of two concepts: watchdog and path-rater. Every node implements a watchdog that constantly monitors the packet forwarding activities of its neighbors and a path-rater rates the transmission reliability of all alternative routes to a particular destination node. The disadvantages of this scheme are that (1) it is only practical for source routing protocols instead of any general routing protocol and (2) collusion between malicious nodes remains an unsolved problem [31].

Rating scheme: In rating scheme the neighbors of any single node collaborate in rating the node, according to how well the node executes the functions requested from it. It strikes a resonant chord on the importance of making selfishness pay. Selfishness is different from maliciousness in the sense that selfishness only aims at saving resources for the node itself by refusing to perform any function requested by the others, such as packet forwarding and not at disrupting the flow of information in the network by intension. The disadvantages of this approach are that (1) how an evaluating node is able to evaluate the result of a function executed by the evaluated node, (2) evaluated node may be able to cheat easily, and (3) the result of the function may require significant overhead to be communicated to the evaluating node [31].

Virtual currency: This scheme introduces a type of selfish node that is called nuglets [32]. To insulate a node's nuglets from illegal manipulation, a tamper resistant security module storing all the relevant IDs, nuglet counter and cryptographic materials, is compulsory. In Packet Purse Model each packet is loaded with nuglets by the source and each forwarding host takes out nuglets for its forwarding services. The disadvantages of this scheme are that : (1) malicious flooding of the network can not be prevented, (2) intermediate nodes are able to take out more nuglets than they are supposed to, and (3) overhead [31].

Route DoS Prevention: It attempts to prevent DoS in the routing layer by cooperation of multiple nodes. It incorporates a mechanism to assure routing security, fairness and robustness targeted to mobile ad hoc networks. The disadvantage of this approach is that misbehaving nodes are not prevented from distributing bogus information on other nodes' behavior and legitimate nodes can be classified as misbehaving nodes [32].

Chapter 3

DIRECTED DIFFUSION

3.1 Directed Diffusion: An Overview

Several schemes have been proposed for routing in WSNs that leverage on sensor network specific characteristics such as application requirements. Directed Diffusion DD [2] is one example of a generic scheme for managing the data communication requirements and thus routing in WSNs. In this chapter, we will present this protocol in detail as this is the particular protocol that is the focus of our study.

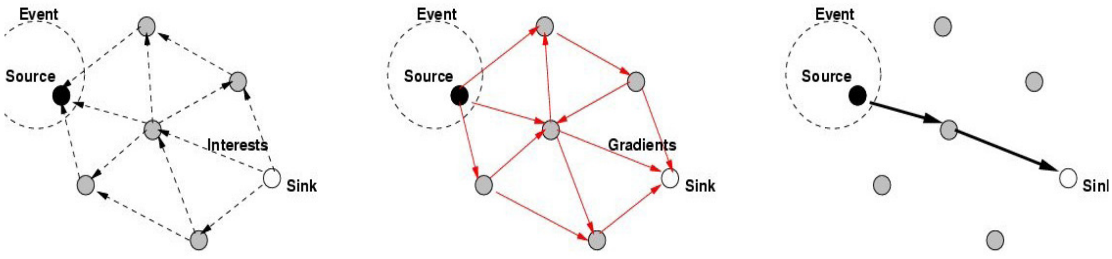
3.2 Directed Diffusion Protocol

3.2.1 Directed Diffusion Algorithm

Directed Diffusion is systematized in five phases (see Figure. 3.1, originally shown in [2]): 1) interest dissemination, 2) gradient setup, 3) low rate data propagation, 4) path reinforcement, and 5) high rate data propagation as it is investigated below.

Interest Dissemination: When a certain sink is interested in collecting data from the nodes in the network, it propagates an interest message. Interest describes a task required to be done by the network. It can be defined by a list of attribute-value pairs such as name of objects, interval, duration, geographical area, etc [33]. For each active task, the sink periodically broadcasts an interest message to each of its neighbors. Every node maintains an interest cache. When a node receives an interest, it first checks to see if the interest exists in its cache. If no matching entry exists, the node creates an interest entry and saves the parameters from the received interest. Then, it re-broadcasts this new interest message to its neighbors.

Gradients Setup: A key feature of Directed Diffusion is that every sensor node can be task-aware—by this we mean that nodes store and interpret interests, rather than simply forwarding them along. Each sensor node that receives an interest memorizes which neighbor or neighbors sent it that interest. To each such neighbor, it sets up a gradient. A gradient represents both the direction towards which data matching an interest flows, and the status of that demand (whether it is active or inactive). This process continues until gradients are setup from the sources back to the sink or Base Station (BS).



(a) Interest propagation

(b) Initial gradients set up

(c) Data delivery along reinforced path

Figure 3.1: Phases of Directed Diffusion operation

The strength of such a gradient can be adapted, which may result in a different amount of information being redirected to each neighbor. Various metrics such as the node's energy level, its communication capability, and its position within the network can be used. Each gradient is related to the attribute it has been set up for. As the gradient setup phase for a certain interest is complete, only a single path for each source is reinforced and used to route packets towards the sink (path reinforcement and forwarding).

Low-Rate Data Propagation (Exploratory Data): In this phase, when a sensor node detects a target, it searches its interest cache for a matching interest entry. If matching entry exists, the node sends low-rate data to the nodes for which it has a gradient. A node that receives a data message from its neighbor attempts to find a matching interest entry in its cache. If no match exists, the data message is simply dropped. If a match exists, the node adds the received message to the data cache and resends it to its neighbors.

Path Selection and Reinforcement Propagation: After the low-rate data reaches the sink or BS, the sink selects and reinforces one particular (e.g., the best) path in order to draw down higher quality events (e.g., higher-rate data).

High-Rate Data Propagation (Real Data): In this phase, the source node computes the highest requested event rate among all its outgoing gradients and sends them to its neighbors. The node which receives the message examines the matching interest entry's gradient list. If there is a lower data rate than the received data rate, it may down-convert the data to the appropriate gradient.

Data Aggregation is performed when data is forwarded to the sink by means of proper methods, which can be selected according to application requirements. The reinforced paths must be periodically refreshed by the sink and this can be expensive in case of dynamic topologies. A tradeoff, depending on the network dynamics, is involved between the frequency of the gradient setup (i.e., energy expenditure) and the achieved performance. A valuable feature of Directed Diffusion consists of the local interaction among nodes in setting up gradients and reinforcing paths. This allows for increased efficiency as there is no need to spread the complete network topology to all nodes in the network.

3.2.2 Differences between DD and Traditional IP Protocols

Directed Diffusion is clearly related to traditional network data routing algorithms, but there are several key features which differ from traditional networking [34].

First, Directed Diffusion is data-centric: The basic scheme in Directed Diffusion proposes the naming of data as opposed to naming sources and destinations of data. Data is requested by name as an "interest" in the network. The request dissemination sets up "gradients" so that the named data or events can be drawn.

Second, all communication in diffusion is neighbor-to-neighbor or hop-by-hop, unlike the traditional data networks with end-to-end communication. In other words, every node is an 'end' in a sensor network.

Third, there are no routers in a sensor network. Each sensor network can interpret data and interest message. This design choice is justified by the task-specificity of sensor networks. Sensor networks are not general purpose communication network.

Forth, sensor nodes do not need to have globally unique address. Nodes, however, do need to distinguish between neighbors.

Finally, in an IP-based sensor network, for example, sensor data collection and processing might be performed by a collection of specialized servers which may, in general, be far removed from the sensed phenomena. In our sensor network, because every node can cache, aggregate, and more generally, process message, it is possible to perform all such tasks.

3.2.3 Advantages of Directed Diffusion

It has been proved that Direction Diffusion has noticeably better energy efficiency, especially in highly dynamic network. This is due to the transmission of data from neighbor to neighbor, thus no data is propagated across the network. For some sensor fields, its dissipated energy is only 60% that of omniscient multicast. Moreover, every delivery has less than 20% additional average delay [2]. Furthermore, the application specific data aggregation in Directed Diffusion shows the benefit of in-network processing. An experiment comparing traffic with and without suppression has proved that suppression is able to reduce traffic. Therefore, it can reduce the bandwidth needed for sensor networks [2]. Also, Directed Diffusion is a robust dissemination in dynamic sensor networks, while at the same time minimizing the per-node configuration that is characteristic of today's networks.

3.2.4 Limitations and Disadvantages of Directed Diffusion

There is limited memory storage for data caching inside the sensor node. Therefore, data aggregation may be affected [35]. In addition, the cost of attribute matching is linear with the number of elements [33].

3.3 Threats, Security Faults, and Vulnerabilities of Directed Diffusion

As a sensory network protocol, Directed Diffusion is subject to many threats and risks as discussed in previous chapters. However, in what follows we are interested in identifying the vulnerabilities of DD due to its infrastructure architectural design (for example, its special control signals).

3.3.1 Vulnerabilities of Unsecure DD

- *Sink could be any arbitrary node:* As a characteristic design, Directed Diffusion allows the user to post queries at any arbitrary sensor node (called the sink). The sink then floods the network with the query. After some time, sensor nodes start sending their aggregated data towards the sink. The sink gives the data to the user. In this case, to prevent the adversary from querying the sensor network, an access control mechanism should be built into each sensor node. But even if all communication between the sensors is properly encrypted and authenticated, and some access control mechanism is built into each sensor node, if the adversary can disable the access control mechanism on a single sensor node, for example by capturing it, he would be able to query the entire sensor network. This single sensor, acting as a new sink, will build a secure authenticated channel to other sensor nodes. This happens because any arbitrary sensor is authorized to act on behalf of the user [37].
- *No Information about Sink ID in Interest Packet*: this design feature of DD interest packet makes spoofing interest possible and even simple.
- *No Information about Source ID in Data Packet:* This enables any node in the network to forge data.
- *Negative Reinforcement Signal:* As negative reinforcement signal is used for path pruning as either a punishment technique for the bad route or a prevention technique of loops. Negative reinforcement could be utilized by an attacker to suppress all flow along sink to source [18]. This is because any node that receives a negative signal (by attacker here) has to negatively reinforce its upper stream node in its turn.

3.3.2 Vulnerabilities of Secure DD

LEAP and LKHW are two different versions of securing techniques presented to secure DD in [38] and [39], respectively. Both of them consider using a single symmetric key for authenticated querying and therefore, are not resistant against impersonation attacks which are possible in case of node capture.

3.4 Previous Work Concerning Directed Diffusion

3.4.1 Previous Attacks against Directed Diffusion

Although a large body of literatures dealt with Directed Diffusion vulnerabilities, the vast majority of such work was devoted to theoretically discuss DD security and the possible attacks threat with no implementations of these attacks as it was the case in [40] and [41] where both papers investigate different misuse actions manipulated to attack AODV and TORA, respectively, to achieve certain attack objectives. In this subsection, we review this work dedicated to security and threats related to DD:

In [18], security in wireless sensor networks has been proposed, the authors present general classes of attacks, and analyze the security of nearly all the currently documented sensor routing protocols including DD. However, this work may be considered as an argument of DD security rather than a real simulation of an attack on DD based sensory network.

Similarly in [38], taxonomy of possible threats to DD is viewed. Some of these attacks are cloning attack, flow suppression, path influence, selective forwarding, and node inclusion/exclusion.

In his paper, Kalambour [42] addresses some of the security issues for routing in sensor networks by taking an example of the Directed Diffusion protocol for analysis of the attacks and general possible countermeasures. He classified the possible attacks on Directed Diffusion protocol under three categories: 1) Denial of Service attacks that has two forms to achieve either by jamming or spoofing negative reinforcement, 2) Modification and spoofing of routing information in which the attacker sends spoofed events at a high data rate to the sink node or base station in order to successfully being able to include itself in the path of the base station and observes all packets sent to the base station, and 3) Dropping or selective forwarding of data.

[43] Shows the vulnerability of DD to sinkhole attack where the attacker attracts network traffic by forging or replaying routing messages through compromised nodes. Subsequently, the attracted traffic is used to misuse the network by selective forwarding, denial of service, or any other attack goal.

In [44], a new attack has been introduced as an "Interest Cache Poisoning Attack" which reflects the vulnerability of data centric approaches in WSNs. The basic idea in this attack relies on the fact that interest cache has limited size, and if the cache is full, and a new interest is received, it will replace the oldest entry. Then, the attack injects fabricated interest packets to replace benign entries in the cache, and when the requested data arrives, it will match no interest in the cache leading it to be dropped.

3.4.2 Securing Directed Diffusion

Directed diffusion routing protocol was proposed by Intanagonwiwat et al. without considering security issues [2]. Subsequent works have focused on aspects of security and mitigating attacks unique to DD.

- *Extended Directed Diffusion EDD using LKHW:* Pietro et al. proposed an extension of Directed Diffusion protocol which provides secure multicasting in [39]. The extended scheme, Logical Key Hierarchy for WSNs (LKHW), provides robustness in routing and security and supports both backward and forward secrecy for sensor join and leave operations. However, it does not provide data authentication.
- *Secure Directed Diffusion using LEAP:* where an existing key management protocol called LEAP has been modified and integrated into the Directed Diffusion [38]. Each node has to store three types of keys: one individual key, one group key and d number of pair-wise keys, where d is the number of neighbors of a node; if there are more neighbors for a node then the node has to store more keys, thus increasing the memory requirement. In the case of negative reinforcement, this algorithm differs little from Directed Diffusion. Each packet contains the information about remaining battery power information and the RSSI value of a node. This DD version could be used for applications which require message authentication and message confidentiality.

- *Secure Directed Diffusion SDD [45]:* SDD is a secure version of DD which includes the same original phases. However, in the first interest propagation step, the TESLA protocol [14] is used to ensure that the interest is from sink D.
- *Secure Diffusion [46]:* In Secure Diffusion, each node keeps only a few localized keys for both neighbor authentication and node-to-sink data authentication. Based on data authenticity and quality, the sink reinforces a high-quality path, and assists the intermediate nodes to select neighbors for reinforcements using local rules. Secure Diffusion ensures the delivery of authentic sensing data to the user, while quarantining the malicious traffic injected by the compromised nodes to their local neighborhood.

3.4.3 Mitigating Attacks of DD

The sensor network commonly uses cryptography for security against unauthorized nodes. However, cryptography can only protect the network against external nodes. In addition, it does not solve the problem of compromised nodes, physical destruction, and DoS type attacks. Therefore, the sensor network requires intrusion detection scheme. Existing researches related to DD intrusion detection are introduced as follows [47].

Fuzzy Logic Anomaly Detection Scheme (FLADS)

In FLADS, DoS type attacks that compromise the Directed Diffusion are focused on, where an adversary's purpose is to cause false alarms or deplete resources of sensor nodes. As a result, network lifetime and availability are considerably reduced. Then, FLADS is designed to protect against malicious message injection attacks and resource depletion attacks launched by the adversary. The FLADS uses factors such as the node energy level, neighbor nodes list, message transmission rate, and error rate in the transmission, in order to monitor abnormal sensor node behavior. The Base Station, BS, or Master Node, MN, generates a detection value to determine whether the attacks exist. The detection value is determined by a fuzzy logic controller, by consideration of the four factors. In order to archive this value, the base station BS or master node MN collect the inform message about the factors from neighbor nodes. The MN creates a new advertisement (ADV) message for notification of its location, and the sensor node creates a new inform message for reporting abnormal behaviors.

Chapter 4

THE PROPOSED DOS ATTACK AGAINST DD

4.1 Background

The key function of sensor networks is to sense some environmental variables and send readings periodically to a base station or send readings whenever someone demands them. Denial of Service (DoS) attack prevents the normal use of communication facilities. In sensor network routing, DoS attacks can be classified into two categories: DoS attack on routing traffic and DoS attack on data traffic. An attacker can launch DoS attacks against a network by disseminating false routing information so that established routes for data traffic transmissions are invalid. An attacker can also launch DoS attacks on traffic by ejecting a significant amount of traffic into the network to clog the network. Both types of attacks might be used to consume valuable network resources such as bandwidth, or to consume node resources such as memory or computation power.

In our work, we tackle the two approaches to form DoS attacks. We propose two types of attacks: The first attack targets routing information by exploiting the vulnerability of Directed Diffusion control signaling to stop providing the sink with requested data and we called it On-Off Reinforcement Swap attack. The second attack, Swam Flooding Attack, refers to traffic injection; two shapes of this attack are discussed (Bee and Ant). This chapter is a detailed explanation of these two attacks and all their related modes.

4.2 System Model and Node Characteristics

We consider a large-scale wireless sensor network in which a massive number of wireless sensor nodes are randomly distributed in the target area. Directed Diffusion is the underlying protocol. The network consists of a large number of sensor nodes such as MICA2 sensors. Every sensor node has limited capabilities in terms of computation, storage, and wireless communication. The sensor nodes operate on non-renewable batteries; once a node exhausts its battery it is considered to be dead. We assume that the sensors are physically insecure, since the physical access to the motes is probabilistically possible in hostile environments.

The user interacts with the network through a data collection unit, called a sink. A sink or base station could be any arbitrary sensor node that can inject quires (interests) to propagate along the network. The queries may be optimized or otherwise processed at the place of injection and then they are disseminated in the sensor network using multi-hop communication according to some query processing mechanism. Sensor nodes whose sensing results match the query disseminate data reports back to the sink over potentially multi-hop wireless links.

The sensor nodes are static since they do not move once deployed. The monitoring task typically requires each node to be aware of its geographic location to tag the sensing data. Such location-awareness can be achieved through either GPS or a localization protocol. We assume that each node can obtain its location within certain accuracy after it is deployed. Figure 4.1 represents our network model.

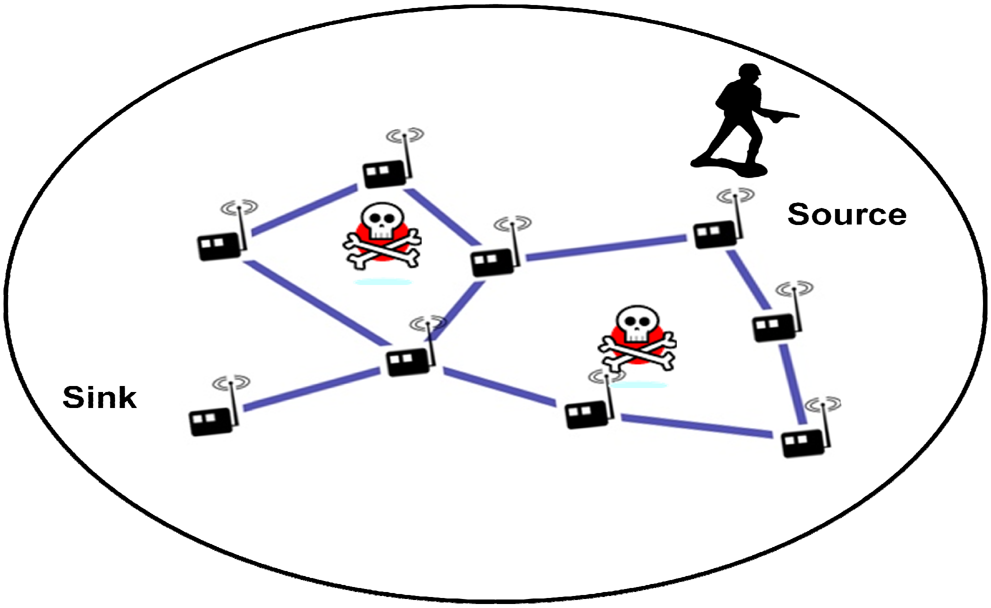


Figure 4.1: Wireless sensor network for our system model

4.3 Design Considerations

The question that we have sought to answer is under what circumstances our DoS attack might be effective. Clearly if we want to deeply degrade the network performance upon starting an attack, we have to attain the following properties in our design:

- *Easy to implement, difficult to prevent, hard to detect.*
- *Simple,* we mean situations in which attackers do not adapt their actions to react to changing values of network performance metrics or to exploit specific protocols executed in the network.
- *Explore Parameter Space of the Attack:* discover what combination of parameter settings in the attack models produces maximal damage on the performance of the network.

4.4 Attack Goals

To successfully attack the network, our model has three goals: (1) Compromise some of legitimate sensors and modify its regular code into the malicious one to build our attacker, (2) The number of these captured nodes has to be sufficient enough to make the required difference in the network performance. And (3) they should be well distributed and organized in the network grid to achieve maximal damage. This sufficient number of attackers with attacking code is now ready to participate in the network. If they can make attempt to take part in the network, attack process is considered an easy operation. Once the system has been compromised, the attackers can redistribute themselves to begin their attack. The compromised system will generate the signals as planned to bring down the targeted network. We formulate these three phases as follows:

4.4.1 Node Compromise: Since sensor nodes are not equipped with tamper-proof or tamper-resistant hardware, any physical attacker would be able to actually compromise a node and download the adapted code. The compromised node becomes a malicious insider where it can perform all the attacks that an outsider can. The malicious insiders can attack the network by spoofing or injecting bogus information. The significance of compromising original legitimate nodes after their deployment over just deploying similar adversarial sensor nodes may not be clear in insecure networks. However, it is more valuable in authenticated environments as it results in possessing node's cryptographic information required for it to be authenticated by other nodes in the network, but exhibits malicious behavior. Moreover, if malicious insiders cooperate and share their keys, each insider may generate any message appearing to originate from any of the compromised nodes. Alternatively, one node to be captured is enough as its key could be used by other attackers. In [12], the authors demonstrate how to extract cryptographic keys from a sensor node using a JTAG programmer interface in a matter of seconds. Although the use of more expensive tamper resistance hardware could be a solution to node compromise problem, this solution would increase the cost per sensor considerably, thus ruling out deployment of sensor networks with thousands of nodes.

4.4.2 Number of Attackers

We need to formulate an appropriate relation to calculate the number of attackers based on number of legitimate sensor nodes, n, transmission range of individual sensors, r, and the deployment area, A. Our work has been influenced by a variety of other research efforts. This part of the design relates to topology controls where it has been a great deal of work in its area. It is important to mention that, though many literatures discuss massive types of attacks, all of them inject the number of attackers randomly without calculating it based on network parameters.

Hierarchal algorithms intensively present different formulas in order to divide the network into cells. In [49], the authors have adapted a simple formula and used it in their paper to partition the network into k clusters, assuming that the network area, A, is known and n nodes are uniformly distributed in the field. Using these two assumptions, the number of cells, k, can be computed by using A and r, the transmission range of the individual sensor node, by the relation:

$$k = \left\lceil \frac{A}{\Pi \times r^2} \right\rceil \tag{4.1}$$

Another method to compute the optimal number of cells in a sensor network was presented in [50] where the optimal number of K cells is obtained using, n, the number of nodes, d is the distance to BS, S_{friss} and $S_{two\text{-}ray}$ are the radio energy parameters. Then, attacker number is given by the rule:

$$k = \left\lceil \frac{\sqrt{n}}{\sqrt{2\Pi}} \sqrt{\frac{s_{friss} \quad amp}{s_{tworay} \; amp}} \frac{M}{d_{toBS}^{2}} \right\rceil \tag{4.2}$$

Although the aforementioned references can divide the network in to relatively reliable number of clusters, which could be used to distribute our attackers, both of these formulas are not satisfactory to us. The former is based on the regular form of sensors. However, the ad-hoc deployment of sensor network makes the field to be deployed in an irregular fashion (e.g. not a linear array, 2-dimensional lattice). More importantly, uniform deployment does not correspond to uniform connectivity owing to unpredictable propagation effects when nodes, and therefore antennae, are close to the ground and other surfaces [51]. While the relation originated in [50] is specific to their scenario as the goal of that study was to minimize energy dissipation, and consequently prolong the network lifetime. In what follows, we aim to find a new approach to divide the network in to multiple zones, in which the attackers are going to be placed, such that the basic principle in network portioning relies on the number of nodes each cell should contains such that the attacker in any cell could communicate with the maximum number of nodes within the same zone.

Our approach: As mentioned before, we want to derive a formula to estimate an appropriate number of attackers that could achieve the success to our swap attack. Swap attack mainly depends on the idea of affecting as more neighboring node as possible. For this reason, our method to find k depends on finding d, the average number of neighbors for every sensor node, using the desired connectivity of the graph discussed in [52]. Assuming p is the probability that a link exists between two sensor nodes, n is the number of network nodes, and d being the expected degree of a node (i.e., the average number of edges connecting that node with its graph neighbors) equals to:

$$d = p(n - 1) \tag{4.3}$$

We need to find out the value of d, the expected degree that a node should have so that a sensor network of n nodes is connected. Random-graph theory helps find this value; G (n, p) is a graph of n nodes and p as defined above. Erdos and Renyi showed that, for monotone properties, there is a value of p such that the property moves from "nonexistent" to "certainly true" in a very large random graph. The function defining p is called the threshold function of a property. Given a desired probability Pc for graph connectivity, the threshold function p is defined by the next formulas presented in [52]:

$$P_c = \lim_{n \to \infty} P_r \left[G(n, p) \text{ is connected } \right] = e^{e^{-c}} \tag{4.4}$$

$$p = \frac{\ln(n)}{n} + \frac{c}{n} \tag{4.5}$$

Therefore, given Pc (the network connecting probability), we can find c (real number), and with the knowledge of n (nodes number), the value of p (probability of connection between two nodes) can be obtained. The expected degree of the node d can easily be estimated using the formula in (4.3) which also represents the average number of sensor nodes that each node can communicate with. Next, k, number of required attackers is just determined as:

$$k = \left\lceil \frac{n}{d} \right\rceil \tag{4.6}$$

Why our approach This formula looks more suitable to cause the success of swap attack than those presented in equations 4.1 and 4.2. As the attacker sends a fake positive/negative reinforcement signal to its neighbors, the effect of the attack will appear as the neighbors themselves will send corresponding positive/negative signals to their surroundings. If the attacker hasn't enough neighbors, its effect would be limited and may be eliminated by the effect made by other legitimate sensors. The basic of our approach relies on the guarantee that every one of our attackers is surrounded by a proper number of neighbors. This number assures that the network is connected and also assures that the effect of the swap attack would be extended to the whole network.

Table 4.1 below contains the number of o attackers calculated from the formulas in equations 4.1, 4.2, and 4.6, for different number of network size n, and network dimensions are 100x100 m^2.

Table 4.1: The estimated number of attackers for different network size using different approaches.

Network Size	[49]	[50]	Our Approach
30	5	0-3	2
50	5	0-4	3
100	5	1-6	6
300	5	1-10	16
500	5	2-13	27
1000	5	3-19	52

Parameters	A = 100 x 100m^2
	r = 25 m
		s_{friss} amp= 10pJ
		$s_{two\ ray}$ amp= 0.0013pJ
		M= 100 m
		75< d_{toBS}<185
			Pc = 0.99999

4.4.3 Attacker’s Distribution

Clearly, if only one node on the border of the network is attacked, the impact on performance metrics that determine the‘health’of the network will be minimal. On the other hand, if the attacked node is a one through which many routes must pass, the impact of the attack will be more noticeable; assuming that attackers are poorly informed, though it is fair to expect that they wouldn’t be able to distinguish a border node from an internal node. For this reason, we assume that every node in the network is equally likely to be attacked. In our model, we divide the whole network into k certain attack zones where k represents the previously estimated number of attackers from formula 4.6. Each such zone shows the zone of attack or the territory of the attack node. Zone size is controlled by the number of nodes in the network which defines a minimum bound on the number of serving attackers to cause the desired effect in degrading network performance characterized by decreasing the throughput at the sink and increasing the corresponding delay of the delivered data.

Figure 4.2 below demonstrates the division of the network in to k attack zones where k equals the number of attackers calculated from equation 4.6. Note that the attackers, represented by red circles, are placed nearly at the center of each attack zone to affect other legitimate sensors, represented by circles in black.

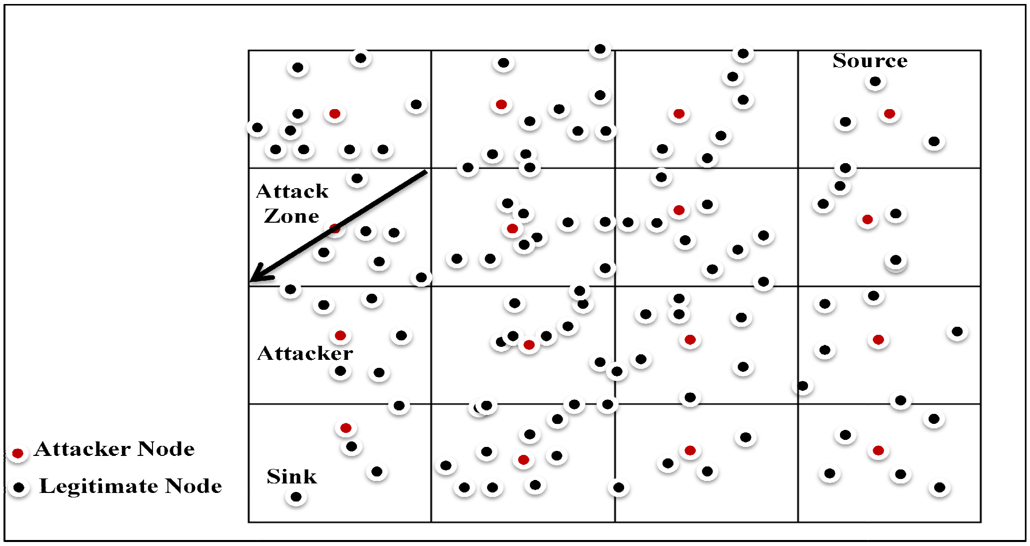


Figure 4.2: Attacker distribution into attack cells throughout the network

4.5 Attack Models

After the malicious modifications of the captured sensors codes, they are placed into their pre-estimated locations. At this level, the attacker can send a request to the normal sensor network to ask for joining the network and whether the protocol has authorization mechanisms or not, the attacker will succeed. This means that our adversary can read and alter those messages transmitted by neighboring nodes to launch a successful denial of service attack. A DoS attack can be perpetrated in a number of ways. Our research is based on two main types of DoS introduced hereunder:

- On-Off Reinforcement Swap Attack: Disruption of configuration information, such as routing information.
- Swarm Flooding Attack: Consumption of computational resources, such as bandwidth, disk space, or processor time.

4.5.1 The First Attack Model: On-Off Reinforcement Swap Attack

Reinforcement Swap Attack

Our attack is based on swapping the rule of routing signaling of Directed Diffusion. As we have explained in chapter 3, DD uses the rules of reinforcement and punishment. On route discovery and establishment, every node monitors its incoming messages, and based on specific parameters it rewards the good path by positive reinforcement, while punishing the bad route by negative reinforcement. Our approach is to swap this rule which means that the good route is excluded and the bad route is included. Although, the spoofing of negative reinforcement alone is enough to clog the data transfer along the network, the inclusion of bad links in the selected route activates nodes that may be so far from the sink, and every node activates another farther node. Most notably, this will consume the power of these sensors, and introduce high delay transfer of data in case that some links still can deliver data to the sink.

Sending fake positive The attacker targets the route establishment in DD operation stages by sending fake positive reinforcement message which replaces negative reinforcement one has to be sent to the neighboring node. When the neighboring node receives this fake positive reinforcement from the attacker, it observes that it already has a gradient toward this reinforcing node but at lower event rate than the rate specified in this interest. In addition we modify the value of the new event rate to be higher than that of any existing gradient. As a result the node must also positively reinforce at least one neighbor.

On selecting the path to be reinforced, the node uses its data cache and local reinforcement rule. In our swapped situation, the attacking node might select the neighbor from which it last received the latest exploratory event matching the interest. Alternatively, it might select all neighbors from which similar exploratory events were recently received. Through this sequence of local interactions, at least one bad data path is established from source to sink.

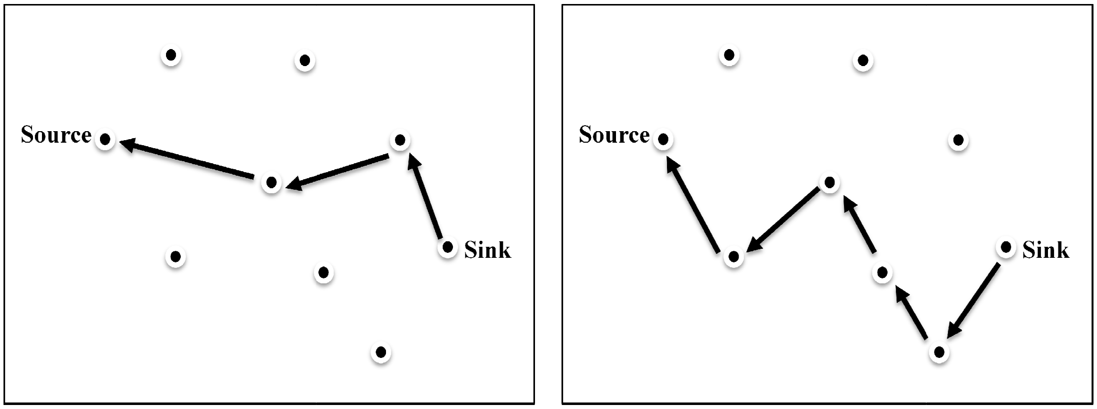
Sending fake negative*:* Here and to explicitly degrade the good path, the attacker sends a negative reinforcement message to the neighbor that delivers the data with the lowest delay. Our rule for explicit fake negative reinforcement is to negatively reinforce the neighbors that have sent events and all of them are new (i.e., those nodes have consistently sent events before their neighbors) within a window of N events or time T. Other alternatives include negatively reinforcing the neighbors that have sent relatively most non-duplicate events to the neighbor that have sent few non-duplicate events.

In this rate-based version of diffusion, the negative reinforcement message is similar to the original interest message except the message type. When the attacker's neighbor receives this negative reinforcement, it degrades its gradient toward this down-link node. Moreover, if now all its gradients are exploratory, it negatively reinforces its neighbors that have been sending data to it (so the effect of the fake negative spread across the neighbors of the attacker's neighbor and so on). This sequence of local interactions ensures the path through the attacker is degraded rapidly, with also increased overhead.

On-Off Reinforcement Swap Attack

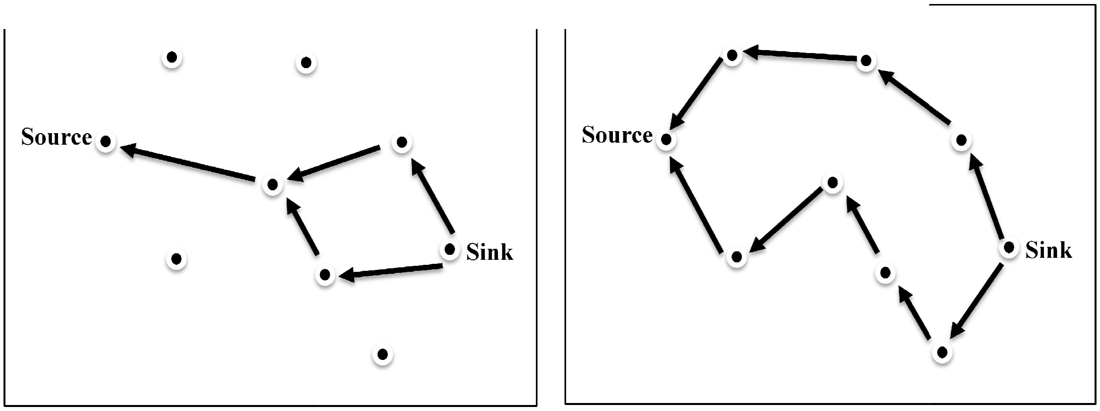
As negative reinforcement attack is a weakness point of Directed Diffusion. Many defense schemes against malicious nodes which are based on trust values of the surrounding nodes can easily detect this type of attacks. In order to enhance our design and strengthen its abilities to be undetectable for longer times, we adapt it to be activated and deactivated rather than continuously swapping the reinforcement signals. In more details, on-off attack means that malicious entities behave well and badly alternatively, hoping that they can remain undetected while causing damage. Moreover, the attacker, when behaves good, may explore the current status of the network since the dynamic properties of DD networks allow the system to recover from the attack and search for alternative paths to deliver the required data. This feedback increases the probabilities of deleting better path along the network. We demonstrate our attack by comparing it with the original DD in two cases. Figure 4.3 shows the first case where only one path is reinforced from sink to source.

Figure 4.3 (b) depicts that the swap attack, unlike original DD in (a), selects the longest path along the network. Also, in Figure 4.4 where it represents the second case in which more than one path is reinforced, our attack reinforces the highest two delay paths.



ration of DD

f DD
d WSN



ration of DD

f DD
ed WSN

Attack Effect

As the goal of the attack may vary from one application to another, the degree of the damage desired by the attack differs also. The damage may be partial and slow in sometimes while being total and fast in other times. For this purpose, we include two modes of our attack as follows classified according to the damage they cause:

<u>Norm Mode</u>: Where the attacker alternates between behaving bad and good in the on/off cycles, respectively. Here the effect of the attack is partial and gradual.

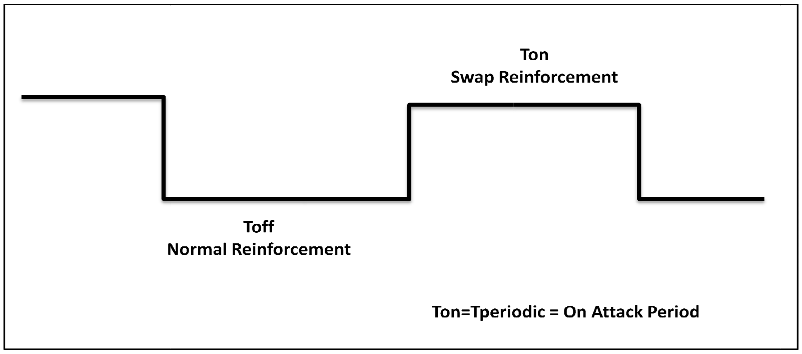
<u>*Halt Mode*</u>: Where the attacker fluctuates between behaving bad and do nothing in the on/off cycles of the attack. Here the effect of the attack is strong and total where the service is denied fast hence the system collapse earlier than the norm mode.

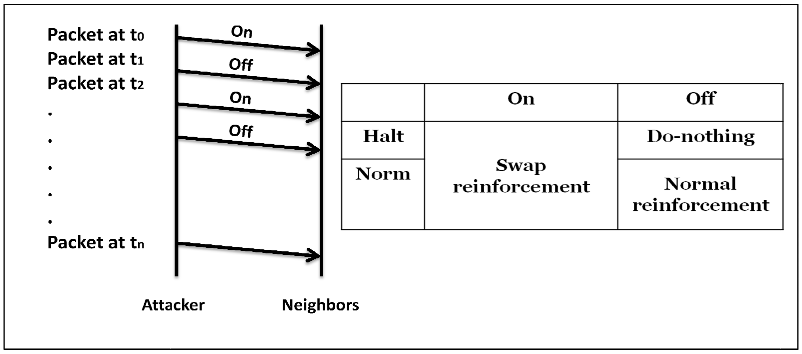
Attack Activation/Deactivation

For the implementation issues of our attack, and to let the attacker exchange between the on and off state, we implement two techniques to switch the attack on and off: Counter and Timer Swap. Both types do what their name implies.

<u>On-Off Timer</u>*:* Where the time is divided into equal frames and the attack host alternates between swapping reinforcement rule for T_{on} seconds and stopping the swap for the next T_{off} seconds. During T_{off} , the attack only activates the norm mode as shown in Figure 4.5.

<u>On-Off Counter</u>***:*** In this type the attack behaves alternatively on receiving a new packet, first it behaves well for the first time, and behaves badly on receiving the next packet. Both norm and halt modes could be applied to counter type (Figure 4.6).





	On	Off
Halt	Swap reinforcement	Do-nothing
Norm		Normal reinforcement

Advantages of On-Off Attack over Continuous Attack

The continued attack may seem attractive at first from the perspective of the attacker because it may make maximal, nonstop disruption on the target network. Such an attack model has its downsides, however.

- *Preserve Attacker's Resources*: in the event that the attack consumes battery power to achieve its goals, the permanent on-state will run down the battery at a steady rate limiting the duration of the attack.
- *Victim Confusing: Attacker* alternates periods of activity and rest. The attacker performs some evil action for a period of time and then stops, leaving the network alone for another period of time. The continuous repetition of this cycle is damaging to network performance because each transition of the attack cycles causes the network protocols to spend time in computation and in communication to reevaluate how traffic should be routed around compromised nodes or communication channels.
- *Expand Life Time of the Attack: On*-Off approach tries hiding a malicious node from the detection mechanism. This is done by taking advantage of the dynamic evolution of trust in the time domain: the behavior of a node keeps changing from good to bad. It would be harder to triangulate the source of the signal allowing the attack to extend for a longer period of time [53].
- *Feedback of the Network State: As* the majority of the ad hoc wireless protocols have their own recovery mechanisms that allow legitimate node to look for alternative routes just after detecting network faults and downs. When the attacker behaves well, that means the attacker is aware of the current status of the network.

Algorithm 1 hereunder gives an idea about how our swap attack works. First, we get the values of the parameters needed for the attack to be properly run. Timer type only works on norm mode where the original reinforcement rule is applied during the off cycle of the attack. As can be seen from the algorithm the counter type can switch between norm and halt mode. The counter attack is triggered on and off based on counter_flag which is set after the receiving of new packets and simply it value is swapped, if it is currently true, it is turned to 1 and vice versa, this allows counter attack to work as it is designed.

Algorithm 1: SWAP ATTACK MODEL *(pseudo-code for an attacker node n)*

```
$k \leftarrow Number\ of\ Attac\ker s$
$Mood \leftarrow Norm / Halt$
$Type \leftarrow Timer / Counter$
$T_{start} \leftarrow Attack\_Start\_Time$
$T_{end} \leftarrow Simulation\_Time$
$T_{ON} \leftarrow Attack\_ON\_Period$
$T_{OFF} \leftarrow Attack\_OFF\_Period$
$Swap\_Fa\lg \leftarrow TRUE$
$Counter\_Flag \leftarrow TRUE$
WHILE ($t > T_{start}$ && $t < T_{end}$) DO:
    WHILE (Swap_Flag==TRUE) DO:
     CASE Type OF
      Timer: {
          Apply Swap Reinforcement Rule for $T_{ON}$ sec
          Do-Nothing for $T_{OFF}$ sec}
      Counter: {
          IF (Counter_Flag = = TRUE) THEN
         { Apply Swap Reinforcement Rule}        ENDIF
          IF ( Counter_Flag = = FALSE) THEN {
          IF ( Mode= =Norm ) THEN
          {Apply Normal Reinforcement Rule}     ENDIF
          ELSEIF (Mode= =Halt) THEN
          {Do-Nothing   }                  END ELSEIF
          }                                ENDIF
          IF (Counter_Flag = =TRUE) THEN{ Counter_Flag = =FALSE)} ENDIF
          IF (Counter_Flag = =FALSE) THEN{ Counter_Flag = =TRUE)} ENDIF
          } ENDCASE
     } ENDWHILE
} ENDWHILE
```

4.5.2 The Second Attack Model: Swarm Flooding Attack

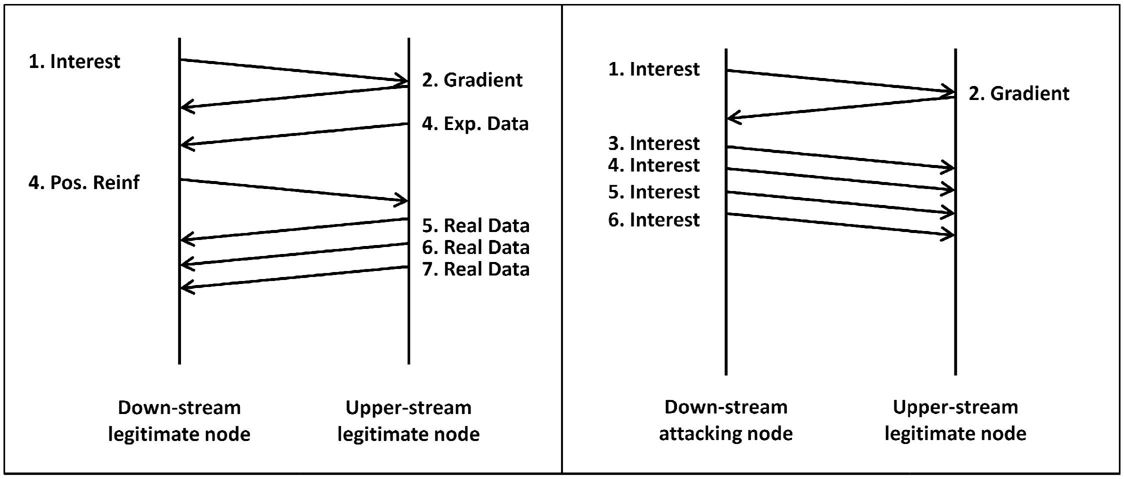
Figure 4.7: Bird swarming

Swarming: The concept of swarming originated from nature. It is a general term that can be applied to any animal that swarms (Figure 4.7). The term applies particularly to insects; Hive or nesting organizations such as ants or bees are the most familiar pattern for swarming [55]. Swarming becomes an interesting research area where the phenomena are utilized to perform useful tasks in all fields. These fields include computing algorithms such as swarming intelligence and swarming optimization. However, most researches into swarming have little to do with swarming attacks in the context of this research. In [56], the concept of swarming attack is presented to perform distributed attacks on a target simultaneously. This method is appealing since the attack comes from so many places; it is difficult to trace the source. Also, once the target is under attack, little can be done to prevent it. Swarm attack is not only characterized by the synchronization among attackers. In addition, it is required that relatively sufficient number of attackers participate to launch the attack.

Flooding: In related framework, there is flooding attack which involves sending large volumes of traffic to a victim system, to congest the victim system's network bandwidth with traffic. The victim system slows down, crashes, or suffers from saturated network bandwidth, preventing access by legitimate users. While in the normal connection between a sink and a source, the five phases of Directed Diffusion are correctly performed. When performing Swarm Flood, the attacker sends several interests but no corresponding sources have the requested data. The connections are hence half-opened consuming server resources. A legitimate sensor tries to connect but all network resources are consumed resulting in a denial of service

For our work, we tend to use and integrate both concepts of swarming and flooding to perform our attack and deny the service to the sink node. We are going to flood the network with massive number of interests to consume network resources. In addition, we will utilize the concept of swarming to test the impact of the synchronization between attackers. Another interesting analogy of swarming in the e physical world is the difference between swarming strategies of Bee and Ant.

Based on this, we introduced two forms of flooding attack namely Bee Swarm Attack and Ant Swarm Attack. Both approaches are inspired from the swarming difference between bees and ants. In all cases, the strategy used to mount an attack is the same. An attack consists of a set of malicious user quires represented by interests, which is inserted into the network and flooded into the whole network until the system is saturated. Figure 4.8 demonstrate the idea of flooding the network with fake interests and compared it with the dissemination of real interests. For the first case (left) all the five phases of DD operation take place. While in the second case when the attackers are present, we can notice that only the first step occurs since no real corresponding events are available for these invalid queries.



s in normal (left)/

1. Bee Swarm Attack

Bee swarming attack is a simple and effective network flooding attack which is inspired by bees' tactic of swarming (Figure 4.9). Bees can only swarm once as stinging results in the stinger's own death. We apply this pattern to attack a Directed

swarm

Diffusion based wireless sensor network where a massive number of interests are injected in the network by the swarm attack simultaneously at the same time. As we mention earlier in this section, swarming implies a sufficient number of malicious nodes to attack the target. As data centric protocol, Directed Diffusion may be most vulnerable to swarm attack. Even with a small number of attackers, the number of interests disseminated by each attacker is another main factor in flooding the network. The attackers of the bee type send different interests with the same data type while in the ant type described later, every swarm sends various data types (available data types are up to 30 in the implementation of network simulator NS-2).

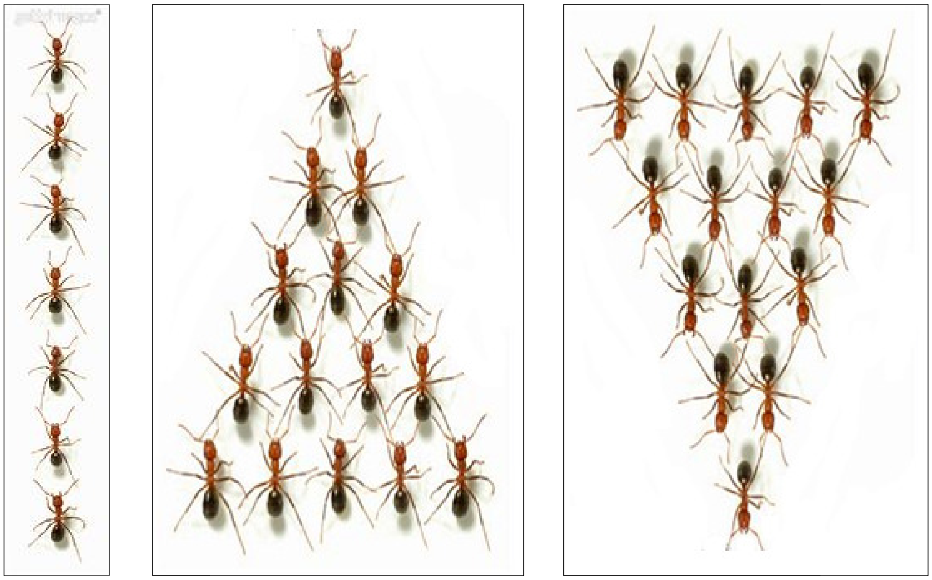
2. Ant Swarm Attack

Ants, on the other hand use the swarm raiding behavior. That is, they move in linear formations, but can shift into swarming mode when it is time to attack. In an ant attack, we explore multiple alternatives and combinations in the terms of attack timing. In our model, we present a new parameter, T_d, which represents the delay between attackers. Inspiring from nature, we consider two variants of ant attack:

(a) Sequential Attack: in which the attackers are injected serially in the network in the terms of entry time and the interest type. This implies that the attackers produce different interest types. As the attackers enter the network earlier, more different interests flooded along the network. Sequential swarming of ant is shown in Figure 4.10 (a).

(b) Forward Hierarchical Attack: which is demonstrated in Figure 4.10 (b), where the attackers enter the network in an increasing swarms or bursts. Each swarm consists of different number of attackers and injects new interests type in to the network. The effect of the attack is expected to be earlier compared to the sequential one.

(c) Reverse Hierarchical Attack: where the shape of ant hierarchical order is reversed to allow larger bursts of attackers to be earlier in the network, Figure 4.10 (c) represents a natural swarming of ants for this technique. The attack effect would be faster than both forward hierarchical and sequential attacks.



(a) Sequential (b) Forward hierarchical (c) Reverse hierarchical

Figure 4.10: Ant swarm

The algorithms of bee, sequential ant, and hierarchical ant are shown next and each of them summarizes the whole process of each attack.

Algorithm 2 presents the simple bee attack where all the k attackers enter the network in sync at $T_{start.}$ The maximum number of interests is propagated throughout the network at the same time and the process continues according to the interest rate.

Algorithm 3 explains the more complicated sequential ant attack in which T_{delay} denotes the time delay between the entries of the attackers in the network. The maximum number of allowed interests is divided equally between the attackers.

Finally, algorithm 4 contains a description of hierarchical attacks which is similar to algorithm 3 except that here the swarm (group of attackers) replaces the concept of the individual attacker in algorithm 3. In other words, the attackers enter the network as increasing or decreasing swarms. For every attacker there is I_{swarm}, the maximum number of interests to be disseminated in the exact entry time specified for every swarm. In addition, the T delay is initiated by the starting of the attack where the first swarm enters the network and is incremented by T_{delay} for each swarm.

Algorithm 2: BEE SWARM ATTACK (pseudo-code for an attacker node n)

```
$k \leftarrow number\ of\ attac\ker s$
$I_{\max} \leftarrow 30$
$T_{sync} \leftarrow Attack - Start\_Time$
$T_{end} \leftarrow Simulation\_Time$
$BEE\_SWARM\_FLAG \leftarrow TRUE$
WHILE (t < $T_{end}$) DO {
   IF (BEE_SWARM_FLAG) = = TRUE THEN {
   FOR (i=0 ; i<k ; i++) {
       FOR (j=0 ; j<$I_{max}$ ; j++) {
          SEND $I_j$ at $T_{sync}$
            }ENDFOR
         }ENDFOR
     }ENDIF
}ENDWHILE
```

Algorithm 3: ANT SEQUENTIAL ATTACK (pseudo-code for an attacker node n)

```
$k \leftarrow number\ of\ attac\ker s$
$I_{\max} \leftarrow 30$
$I_{atakr} \leftarrow 30 / k$
$T_{sync} \leftarrow Attack - Start\_Time$
$T_{delay} \leftarrow Delay - Between - Attac\ker$
$T_{end} \leftarrow Simulation\_Time$
$j \leftarrow 0$
$ANT\_SEQ\_FLAG \leftarrow TRUE$
WHILE (t < $T_{end}$) DO {
   IF (Ant_Seq_ Flag )= =TRUE THEN {
   FOR (i=0 ; i<k ; i++) {
     WHILE ( j < $I_{swarm}$+i x $I_{swarm}$ ) DO{
          SEND $I_j$ at $T_{sync}$
            } ENDWHILE
                 $T_{sync} \leftarrow T_{sync} + T_{delay}$
         } ENDFOR
     }ENDIF
}ENDWHILE
```

Algorithm 4:ANT HERARICHICAL ATTACK (pseudo-code for an attacker node n)

```
$k \leftarrow number\ of\ attac\ker s$
$S \leftarrow number\ of\ swarms$
$I_{\max} \leftarrow 30$
$I_{swarm} \leftarrow 30 / S$
$T_{sync} \leftarrow Attack - Start\_Time$
$T_{delay} \leftarrow Delay - Between - Swarms$
$T_{end} \leftarrow Simulation\_Time$
$j \leftarrow 0$
$ANT\_HERARY\_FLAG \leftarrow TRUE$
WHILE ($t < T_{end}$) DO {
  IF (ANT_HERARY_ FLAG)==TRUE THEN {
  FOR (i=0 ; i<S ; i++) {
    WHILE ( $j < I_{swarm} + i \times I_{swarm}$ ) DO{
        SEND $I_j$ at $T_{sync}$
          } ENDWHILE
              $T_{sync} \leftarrow T_{sync} + T_{delay}$
      } ENDFOR
   }ENDIF
}ENDWHILE
```

4.6 Related Work

Although many attacks have been proposed towards different unsecure WSNs protocols [40] and [41], no real attacks concerning DD and involving real analysis (simulation) has been tackled before. For our swap attack, some theoretical proposal about weak tips of DD concerning sending negative reinforcement to deny the service has been tackled in many literatures like [18] and[42]. However, our work is different than all the previous proposals. First, it investigates more detailed attack concerning degrading more than metric in the network (increasing delay, deplete more energy besides decreasing throughput). Second, it explores distinct applications since it introduces different types and modes of the attack according to the desired degree and effect of the attack on the victim network. For swarm flooding attack, most of existing *flooding* attacks exploit the three-way handshake mechanism in TCP/IP protocol and not applied to networks. Only one work introduced Ad Hoc Flooding Attack [57] briefly attack a network running AODV protocol and doesn’t explore any specifications of the attack.

Only one work [56] theoretically mentions using the concept of swarming in attacking the web servers.

For our work, we exclusively integrate both concepts of swarming and flooding to obtain more than one form of the attacks. In addition, this study explores the parameter space of the attack and specifies the most significant factors that cause more damage to the network.

Chapter 5

SIMULATION AND DISSCUSSION

5.1 Background

This chapter details our simulation model and provides an analysis of simulation results obtained. For our experiments, we have used the Network Simulator (NS-2.32) [3] to simulate a wireless sensor network running the Directed Diffusion routing protocol.

5.2 Simulation Setup

5.2.1 Simulation Tool: The NS-2 simulator is a discrete event-driven network simulator, which is popular with the networking research community [3]. It was developed at the University of California at Berkeley and extended at Carnegie Mellon University, CMU, to simulate wireless networks [4]. These extensions provide a detailed model of the physical and link layer behavior of a wireless network and allow arbitrary movement of nodes within the network. It includes numerous models of common Internet protocols including several newer protocols, such as reliable multicast and TCP selective acknowledgement. Additionally, different levels of configurations are present in NS-2 due to its open source nature, including the capability of creating custom applications and protocols as well as modifying several parameters at different layers.

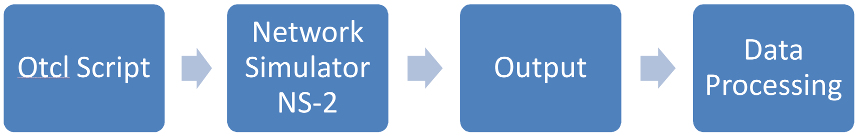


Figure 5.1: Description of simulation approach using NS-2 tool.

The simulator is written in C++, accompanying an OTCL script language based on Tcl/Tk. The researcher defines the network components such as nodes, links, protocols and traffic using the OTCL script i.e. NS-2 uses OTCL as the interface to the user. This script is then used with NS, the simulator, to conduct the desired simulation, and as a result outputs traces at different selective layers. The output data within the trace output files is then filtered and extracted using statistical analysis software like excel/access program. The extracted relevant data is then used to evaluate performance by manipulating various metrics such as delays, throughput, overheads etc.

5.2.2 Simulation Parameters

We emulate the actual network environment including radio propagation model and MAC layer. In our simulations, the physical layer assumes a fixed transmission range model, where two nodes can directly communicate with each other successfully only if they are in each other's transmission range. Simulations are implemented with 1 sink and 1 source. The source is located at the most right region of the simulation area, while the sink is placed at the most left area. This ensures that our results are representative of a long multi-hop path from source to sink. It also permits the introduction of failures at various distances from the source. A 64-byte data event is sent every 0.5s, 36-byte interest every 5s, and 64-byte exploratory data event every 50s. Simulation parameters were chosen in accordance with [2] and listed in Table 5.1.

Table 5.1: Summary of the values of the parameters used in simulation scenarios

Parameter	Value
Simulation Time	1300, 1500 second
Simulation Area	800m x 800m
Number of nodes	30
Number of attackers	4
Transmission Range	250 m
Link Bandwidth	1.6 M bps
Propagation Model	Two-Ray-Ground
Data Link Layer	MAC IEEE-802.11
Routing Type	DIFFUSION/RATE- MYDIFFUSION/RATE
Traffic Type	Diff_Sink - MyDiff_Sink
Tx/Rcv Power	0.66/0.395 J
Ideal/Initial Power	0.035/100 J

5.2.3 Implementation Details

To verify our attacks in Directed Diffusion, we implemented them in NS-2.32. We have implemented two types of attacks through simulations explicitly reinforcement swap and swarm flooding attacks. The Ns-allinone-2.32 simulation software is compiled and run in WinXP-Intel®Core™2Duo CPU-Cygwin-2.573.2.2. Cygwin provides a Linux-like environment under Windows. Diffusion module in NS-2 has two versions, Diffusion and Diffusion3. For our implementation, we use the Diffusion edition programmed by Intanagonwiwat. This version of Diffusion has two types; diffusion/rate and diffusion/prob. Apart from the original Diffusion/Rate routing protocol, another malicious routing protocol named MyDiffusion/Rate is generated during the implementation. Both protocols inherit the same packet format and routing mechanisms. But the send and receive functions of MyDiffusion agent are overwritten with our attacking code. Note that, we add our codes while keeping the original code untouched to allow the malicious entities to alternate between MyDiffusion and Diffusion code during the on and off periods respectively as it was previously discussed in the swap attack.

For all the simulations, we used a tcl program to generate a wireless network of N nodes. The first K nodes represent the attackers who run MyDiffusion codes, while the other N-K nodes correspond to the legitimate sensor nodes running normal version of Diffusion**.**

5.3 Simulation Scenarios

To support different research methods, we have chosen to let the attack work in more than one mode. Each mode has its own advantages for certain scenarios.

Choosing an appropriate simulation scenario to study the performance of routing protocol under attack is an important process. For example, an attack will not be properly evaluated when a simulation scenario is run with a low data rate or if small simulation time is considered. To ensure that a simulation scenario provides an effective platform for testing our attack, we use two main metrics to characterize our simulation scenarios: the throughput and the average delay. In this study, we conduct several models that take the desired values for different variables as inputs (data rate, number of attackers, interest rate, number of interests), and output these metrics (throughput, average delay) to create a simulation scenario that meets the researcher's target values for these two metrics to a close approximation. In this way, we provide several models that researchers can use to construct an optimum attack, which meets their demands on how they choose to deny the service of DD based WSN. To examine the impact of our proposed attacks, we investigate several scenarios under several considerations. We consider the attacks as categorized in Figure 5.2.

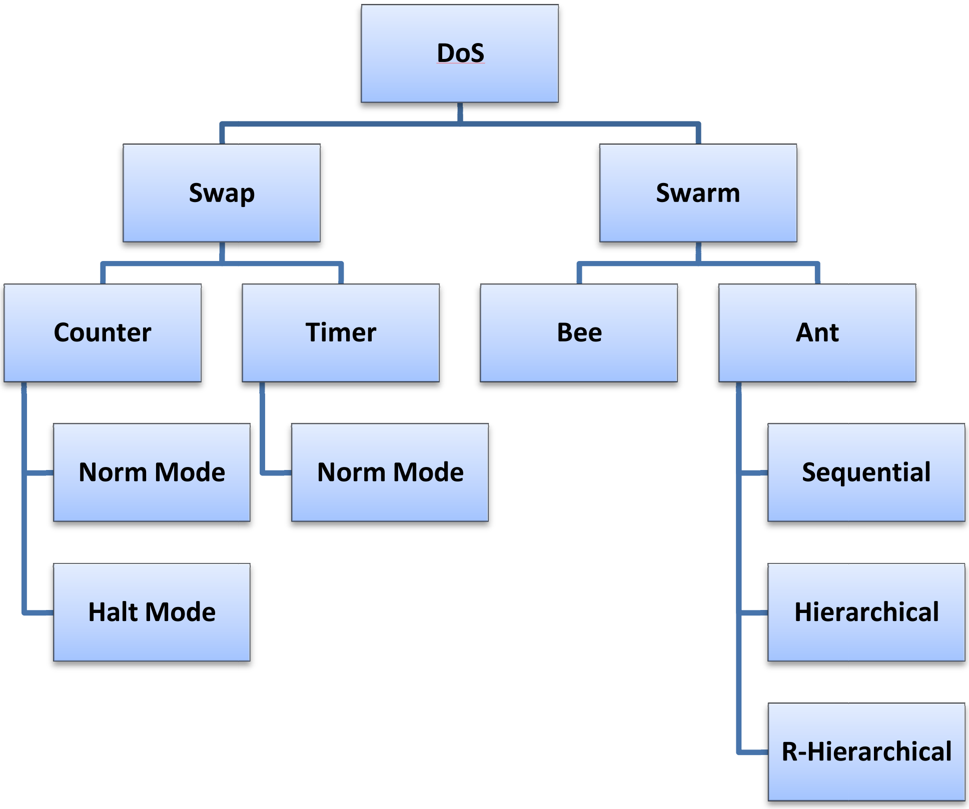


Figure 5.2: Summary of attack simulated models

5.4 Performance Metrics

The metrics are the important determinants of network performance, which would have been used to compare the performance of the proposed schemes in the network with the performance of the original protocol. This study has been done to show that the proposed schemes cause substantial degradation in network performance. We choose two main metrics to evaluate the performance of our proposed schemes namely throughput and average delay. However, throughout this chapter, we will use other metrics to measure the efficiency of our work; in what follows, we define these parameters:

1. **Throughput**: It is the sum of received packets at sink, calculated at every time interval and divided by its length. Throughput shows numbers of packets in every time interval. This metric is the most relevant to our work as it reflects the effectiveness of our attacks in preventing data sent by source to be delivered to the sink as much as possible.
2. **Packet Delivery Ratio**: ratio of the packets delivered to the sink to those generated by the sources.
3. **Average Delay**: Average time difference (in seconds) between the time of the packet receipt at the destination node, and the packet sending time at the source node. This includes all possible delays caused by buffering during route discovery latency, queuing at the interface queue, retransmission delays at the MAC, propagation, and transfer times.
4. **Number of Dropped Packets**: The number of data packets dropped at any given node. This is an important parameter because if the number of dropped packets increases, the throughput would decrease.
5. **Routing Overhead**: This measures the efficiency of the routing protocol. It is defined as the ratio between the total number of routing packets transmitted to data packets, or the number of Control Packets produced per mobile node. Control packets include route requests, replies and error messages.
6. **Deny Time**: The time required by an attacker to deny the service to the sink node; we wish to minimize this value to disrupt the system as fast as possible.
7. **Number of Interest Packets**: The number of interests received by the source node; This is an indicator on how much our attacker is successful not only in affecting the sink node, but also it has an impact on the source node.

5.5 Evaluation and Results

5.5.1 Simulation Results of Counter Reinforcement Swap Attack

We have performed a set of experiments to analyze the effect of our DoS attack that a malicious node may launch against DD based network. The DoS attack we have simulated in these experiments is comprised of repeatedly affecting control packets so as not to allow other nodes to successfully forward their data packets through the right routes. In order to numerically evaluate the effects of our attack, two main metrics, the throughput and the average delay, are introduced to measure multiple variants of the network as would be explained in this section.

Effect of Reinforcement Swap Anatomy on Network Throughput while Changing Data Rate

In the design of our first attack, we let the attacker node periodically changes its rules in neighbor reinforcement. The first rule is to swap positive reinforcement with negative signal which results in the deactivation of the good route. In contrast, the negative reinforcement swap rule aims to activate more bad routes by sending positive reinforcement instead of negative. We prefer to make our attackers alternating between swapping both rules during simulation time. In order to justify our preference, we use the simulation to investigate the effect of each swap individually. According to our design, we break up the attack to its components. First, we study the effect of swapping positive and negative reinforcement individually to realize the value of integrating them together in our attack model.

The results are presented in Figure 5.3 which demonstrates that the throughput of the network varies when the data rate of the source changes in each one of the mentioned cases. In the three scenarios, revealed in Figure 5.3, the throughput at the sink node increases for relatively small data rate, and for values more than 50 packets per second, the throughput starts to decrease as the system is saturated due to the heavy traffic generated by system nodes which causes the network to be congested.

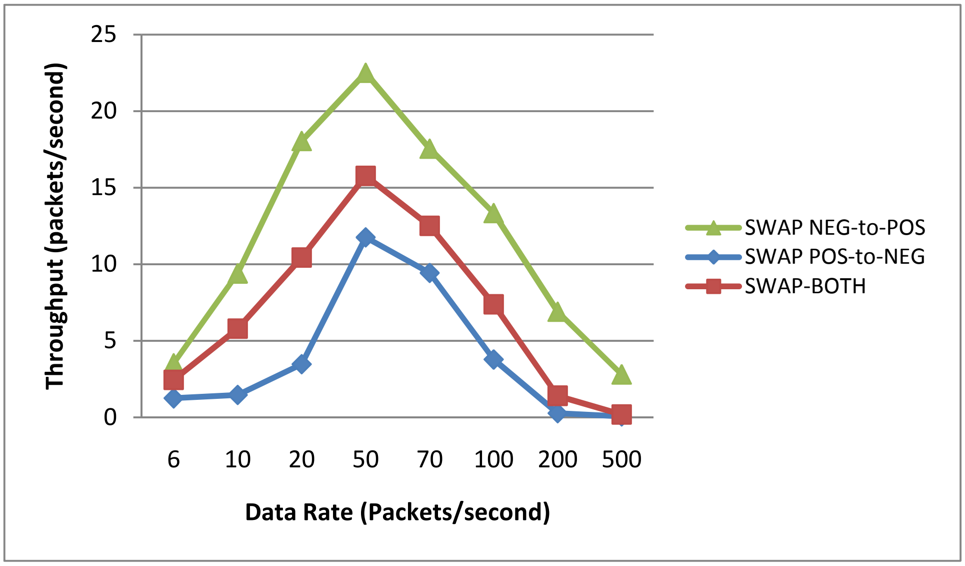


Figure 5.3: Effect of reinforcement swap anatomy on network throughput

The curve of swapping positive to negative has the least values of throughput as it prunes the active routes by negatively reinforcing high data rate gradients in the network. On the other hand, swapping negative to positive has the uppermost values due to the activation of more bad routes positively reinforced by the attacker. The alternative integration between both mechanisms generates a modest curve which is a midway between the two curves. As one can wonder why to make things harder while simply we can select swapping positive to negative scenario in which the lowermost throughput is achieved. We answer this question in Figure 5.4 below.

The content of Figure 5.4 is the same as Figure 5.3 but the y-access represents the average delay instead of throughput. As noticed, the figure illustrates that network delay has an opposite behavior of the throughput given in Figure 5.3. Swapping negative to positive has the maximum delay while swapping positive to negative has the minimum delay. These results are also valid for small data rate values; however, the figure doesn't scale well for the two margins of data rate values. These results are easily explained by the fact that in the first case, the attacker elects the higher delay path to route the data, while in the second case most of the paths are eliminated and the routing of data is done via limited routes which decreases the resultant delay upon calculation. Together these results show convincingly the benefits of using our alternative integrated swap reinforcement scheme, particularly at moderate-to-high data rates.

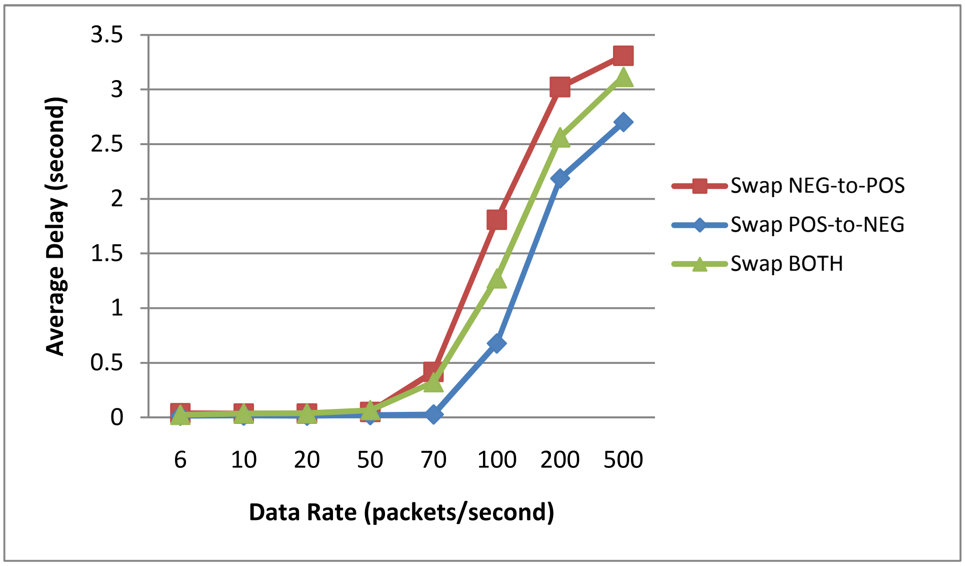


Figure 5.4: Effect of reinforcement swap anatomy on network average delay

Performances of Different Swap Attack Modes while Changing Data Rate:

Figure 5.5 depicts the sink throughput for three DD flows with changing data rate. The curve labeled 'Normal Diffusion' shows flow's throughput in the absence of any attack. Observe that as increasing the data rate, more packets are generated by source, forwarded by intermediate nodes, and delivered to the sink. This explains the rise of the curve when data rate is relatively small. However, as the data rate is getting higher, the data and control signaling increases at a fast pace and directs the system toward congestion. This congestion is represented in the figure as the sharp shrink in the throughput after the data rate exceeds 50. This explanation is also valid for the other two curves. However, the effect of swap attack shifts the two curves down the original Directed Diffusion one. We observe that the halt mode generates the minimum throughput as the attackers swap between bad behavior and halt. On the other hand, the good period on the norm mode of the attack gives the system the opportunity to recover from the bad period and discover new healthy routes.

As a result, there is a modest number of data packets delivered to the sink during the norm mode.

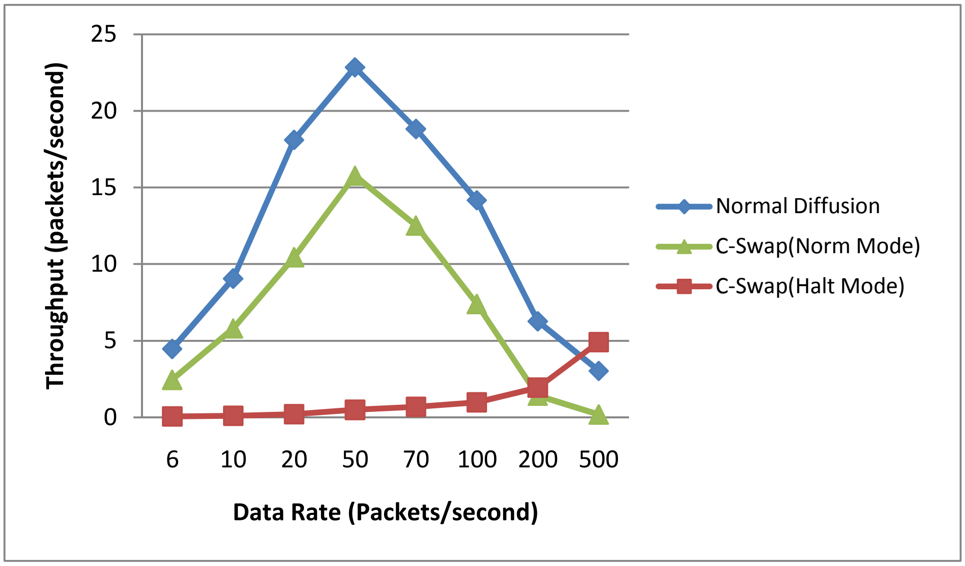


Figure 5.5: Effect of different swap attack modes on throughput while changing data rate

We next performed another analysis to measure another essential metric which is the average delay. Figure 5.6 reveals that our attack method causes more damage to the sensor network communication requiring more time for the network to deliver the requested data to the sink. This is a result of swapping negative to positive seeing that the longer routes (larger delay) have been activated by the attacker.

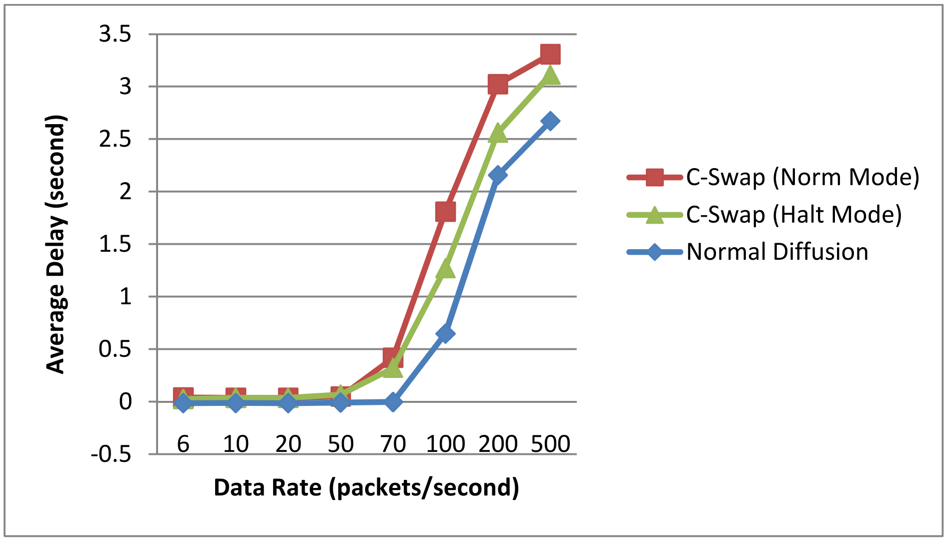


Figure 5.6 Effect of different swap attack modes on average delay while changing data rate

In the same context and with varying data rate, we measure the value of received interests by source to further understand the behavior of the system, and we plot the results in Figure 5.7. As the figure indicates, there has been a decline in the received interests at the source.

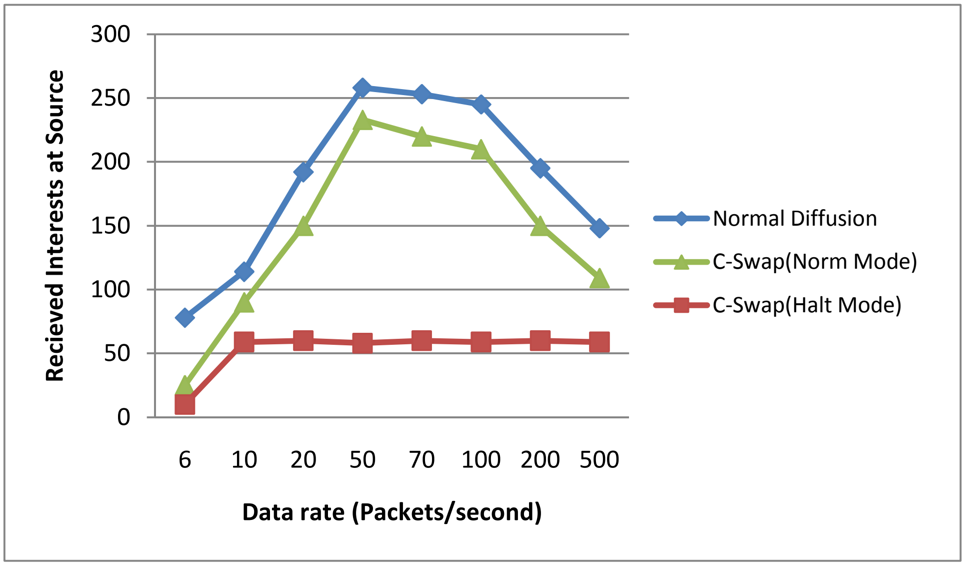


Figure 5.7: Effect of different swap attack modes on received interests at source while changing data rate

<u>Performance of Different Swap Attack Modes over Time:</u>

We previously observed that changing data rate has a strong effect on both throughput and delay. In this simulation, we perform similar analysis to investigate the performance over time. Figure 5.8 systematically compares the throughput of conventional DD network with and without our attack as a function of time. Note that throughput goes down when the intruder starts attacking the network. The throughput is nearly 9 in the normal operation of Directed Diffusion without attack and many packets get to the destination node. However, the throughput is rapidly declined from 9 to 0.2 in the halt mode attack. In other words, most packets can not reach their target and those packets are discarded by nodes for network congestion, and the network can not bear the attack anymore and the performance goes down quickly. It implies that Reinforcement Swap Attack can result in denial of service of whole network. Interestingly, the network seems to have some recoverability when the attacker behaves normally in the off cycle of the attack, the performance becomes better after 400 seconds period and the throughput is averaged to be 5.22. Note that for the analysis of this research we choose the throughput not the packet delivery ratio to compare the performance of the existing DD protocol with and without our attacks. To explain this, we plot packet delivery ratio as a function of time (Figure 5.9). It is found in our study, as the graph can tell that our attacker can produce no pronounced effects on this metric.

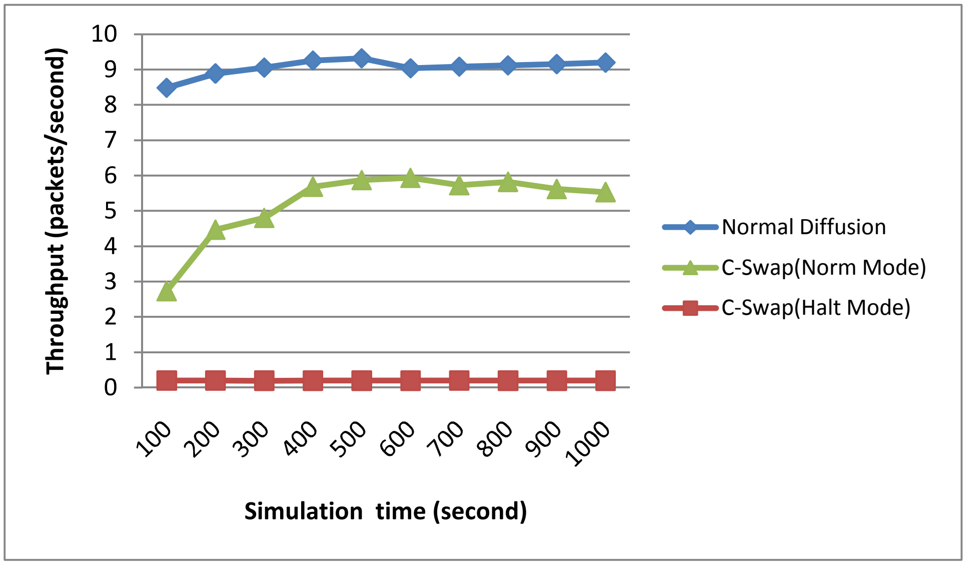


Figure 5.8: Throughput of different swap attack modes over time

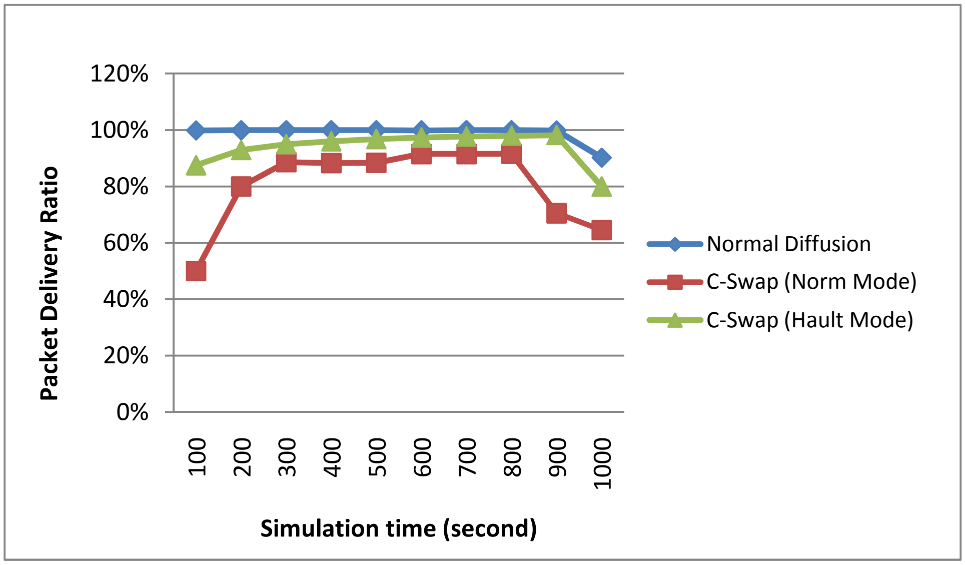


Figure 5.9: Packet delivery ratio of different swap attack modes over time

Packet delivery ratio is almost constant throughout time hence we can't judge the influence of our attacks on the network. The reason behind this is that the number of received packets at the sink decreases as a result of decreasing the number of sent packets by the source. The following graph (Figure 5.10) justifies this drop in the sent data at source node. We found that, during our attack, the number of received interests at source decline as compared to the case when original DD strategy is used for routing.

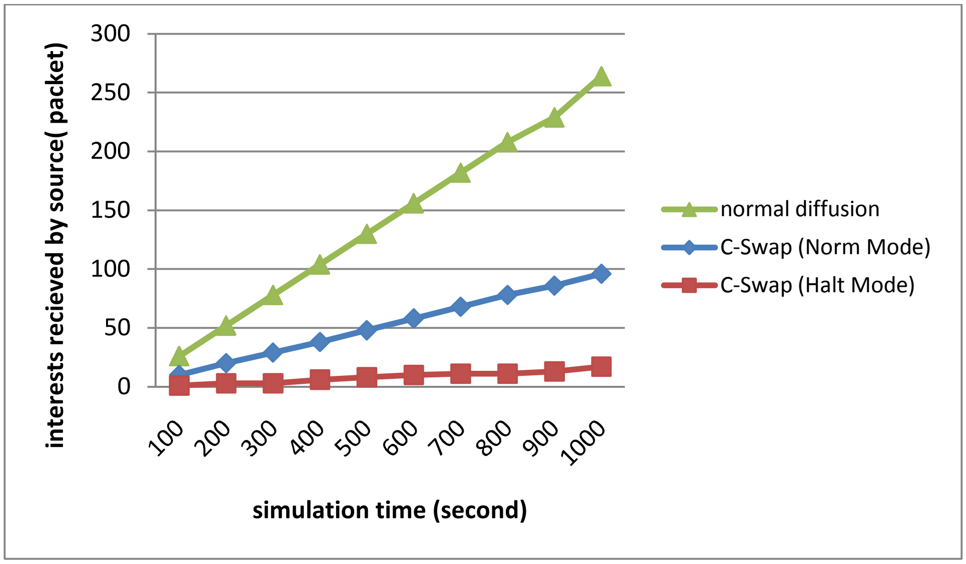


Figure 5.10: Received interests by source for different swap attack modes over time

Figure 5.10 indicates that the number of interests received by source is not constant. This means that the impact of our swap attack is not limited to the sink but also can effectively change the source manners. Typical communication rules reveal that the source continues sending new data packets only if it receives the acknowledgement of the previously sent data. In our case, small number of data is delivered to the sink, consequently not enough acknowledgment is sent to the source which leads to this huge decrease in the sent data.

Finally, we perform other simulations to measure the performance of the network under our attack on routing overhead and dropped packets. The results are shown in Figure 5.11 and Figure 5.12. Both figures share the same characteristics and present the same performance for the three compared cases. The explanation is that, the dropped packets have a proportional relationship with the network signaling. And, as our swap attack mainly intends to degrade the performance via manipulating the route establishing. This manner of the attack guides the network towards less traffic in its two modes compared to original protocol which accordingly produces the shape of both figures. We can see relatively high drop of packets in the three competitive schemes. These high values are due to the MAC layer implementation of this version of Diffusion in NS-2. MAC-802.11 doesn't try to retransmit broadcast packet in case there is a collision and the packet is simply dropped. Coupled with this fact, MAC-802.11 doesn't perform random selection of slots in the contention window before it transmits a packet [70].

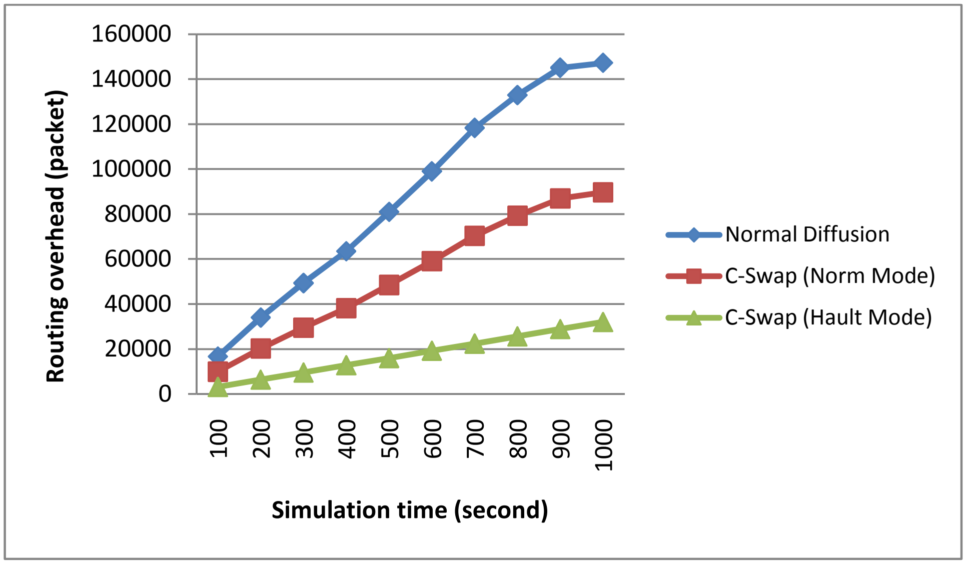


Figure 5.11: Routing overhead of different swap attack modes over time

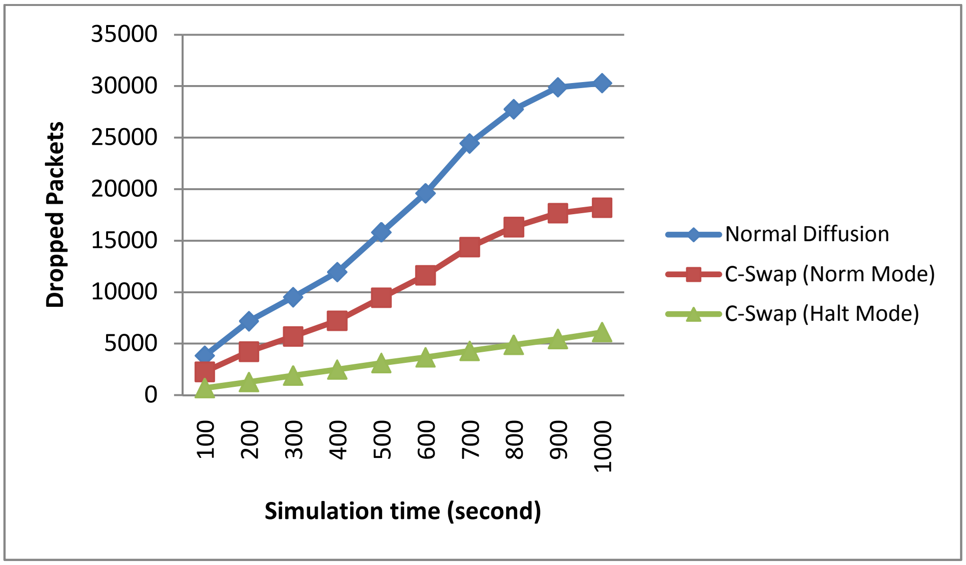


Figure 5.12: Dropped packets of different swap attack modes over time

Performance of Single/Multi Path(s) Norm Mode Attack while Changing Network Size:

We conduct another experiment to test the sensitivity of Directed Diffusion parameters, particularly the reinforcement of single/multiple path(s), toward our swap attack. Figure 5.13 suggests that the protection against DOS attacks varies across different network sizes. As expected, in all cases, the multi-path algorithm provides better protection against DOS attacks than the single path approach. The multipath approach performs far better because the large network nearly always offers a valid redundant second path. The worst performance of the multi-path approach is obtained for small networks in which nodes have few neighbors and few alternate paths (usually only one path) to the base station. In this case, the multi-path approach performs only slightly better than single path routing.

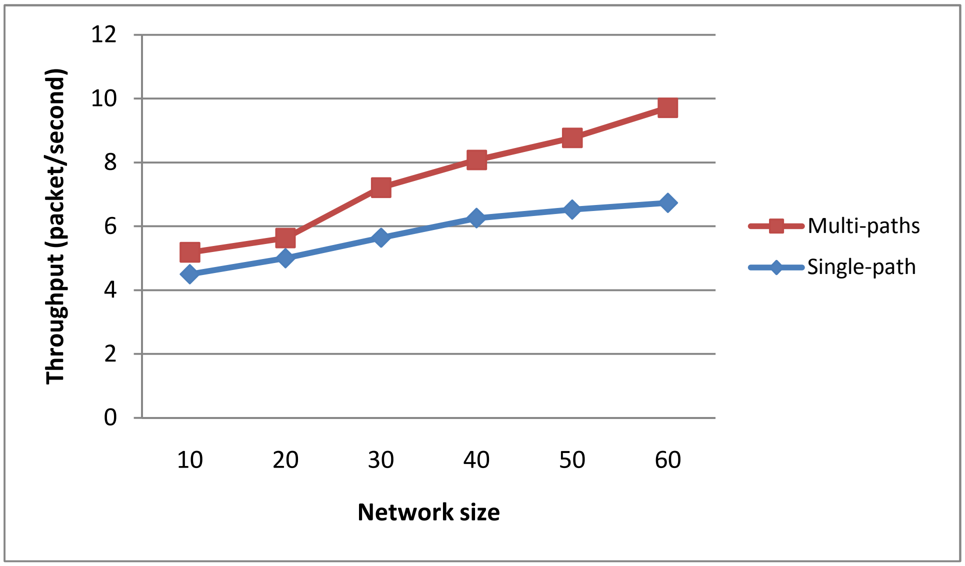


Figure 5.13: Performance of single/multi path(s) norm mode attack while changing network size

Performance of Different Attack Modes when Changing the Number of Attackers

We carried out another simulation to determine the influence of varying the number of attackers on sink throughput. Figure 5.14 illustrates the simulation results with the configurations of up to 10 malicious nodes injected in regular distribution through the network.

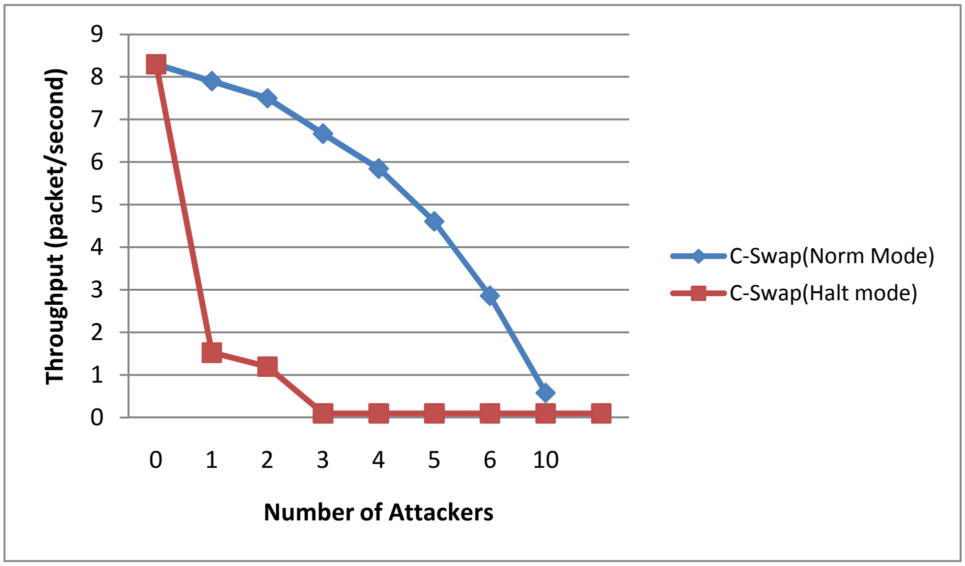


Figure 5.14: Performance of different attack modes when changing the number of attackers

We observe that the performance of both networks has been degraded as the number of attacker increases. The figure also confirms that DD has better robustness to norm mode than halt mode as the throughput is decreased linearly with increasing attackers. For relatively increased number of attackers, they can be distributed and propagated throughout larger space in the network and their impact is more significant which deteriorates network performance.

5.5.2 Simulation Results of Timer Reinforcement Swap Attack

As previously presented in chapter 4, we introduce two approaches to activate and deactivate the swap attack: counter and timer. Thus far, we have considered only counter implementation of the attack; here we study the timer implementation of it.

For the performance analysis of Timer Swap Attack, we introduce another parameter, $T_{periodic}$, the period of on/off cycle consumed by the attacker for acting bad and do nothing in the halt mode or acting bad and good (norm) in the norm mode. Unless mentioned otherwise, the on and off cycles are kept the same length in all cases and we call it the attack period, $T_{periodic}$.

Performance of Timer Swap Attack as Changing the Attack Period

From Figure 5.15, we can see that the protection against DoS attacks varies significantly across different periods of $T_{periodic}$. As expected, in all cases, the very small cycle provides better protection against DoS attacks than the large values. When the cycle equals 0, this case corresponds to the absence of any attack in the network, and with relatively small periods of cycles, the attack is effective as we can notice up to 6 seconds.

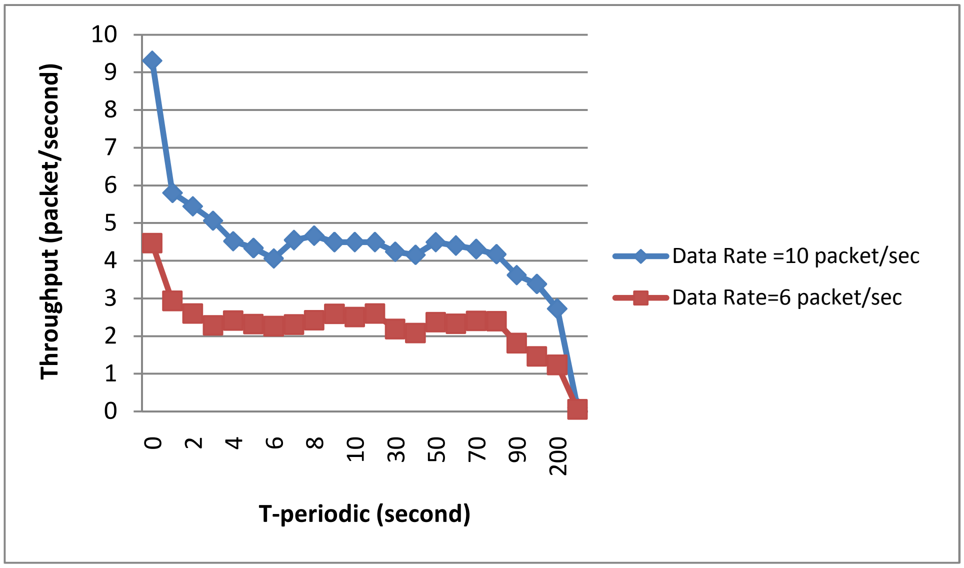


Figure 5.15: Performance of timer swap attack with changing attack period

Then it is observed that for modest values the system is stable and the effect of the attack is limited as the on period consumed by the attacker to behave badly corresponds to similar off cycle which is enough to recover from the effect of the attack. Note that, the performance is evaluated for the norm mode of the attack. As $T_{periodic}$ gets larger, the attack performs far better because the on cycle exceeds the off cycle till the throughput reaches 0 when the attack is continuous.

Performance of Timer Swap Attack as Changing the Data Rate

Next, we consider the relation between data rate and attack period. The experiment is repeated four times with $T_{periodic}$ has the values of 2, 100, 300, and -300 which correspond to no attack scenario. As we can see from Figure 5.16, as attack period increases the attack is more successful as the on cycle increases on the expense of off cycle. Our results indicate that the timer continuous attack has the same behavior of counter halt mode.

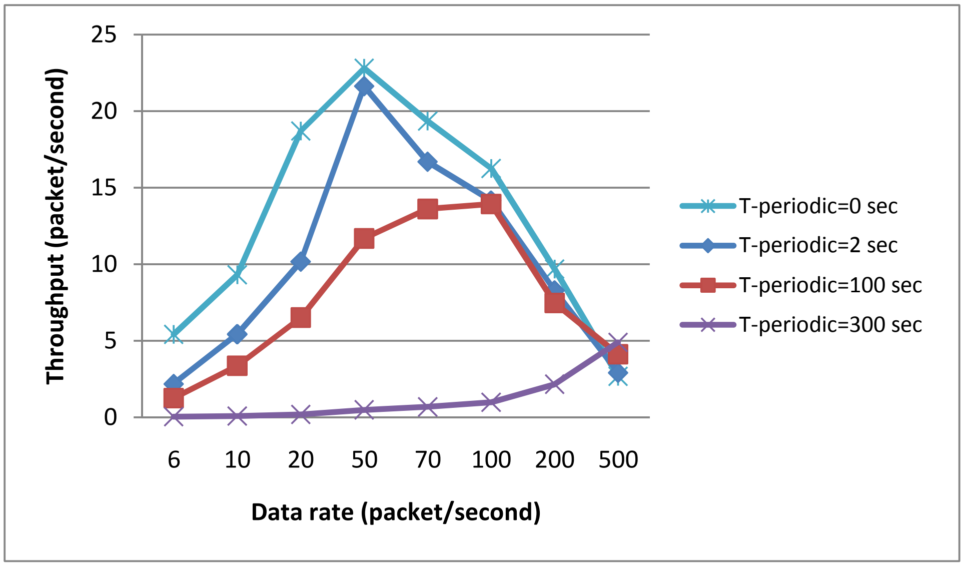


Figure 5.16: Throughput of timer swap attack with changing data rate

For the same parameters of Figure 5.16, Figure 5.17 depicts the performance as a function of average delay. The figure demonstrates that the delay increases when increasing the attack period as more bad routes are reinforced, which caused the delay in receiving the data at the sink. However, the effect of attack period is partial for different periods of the attack.

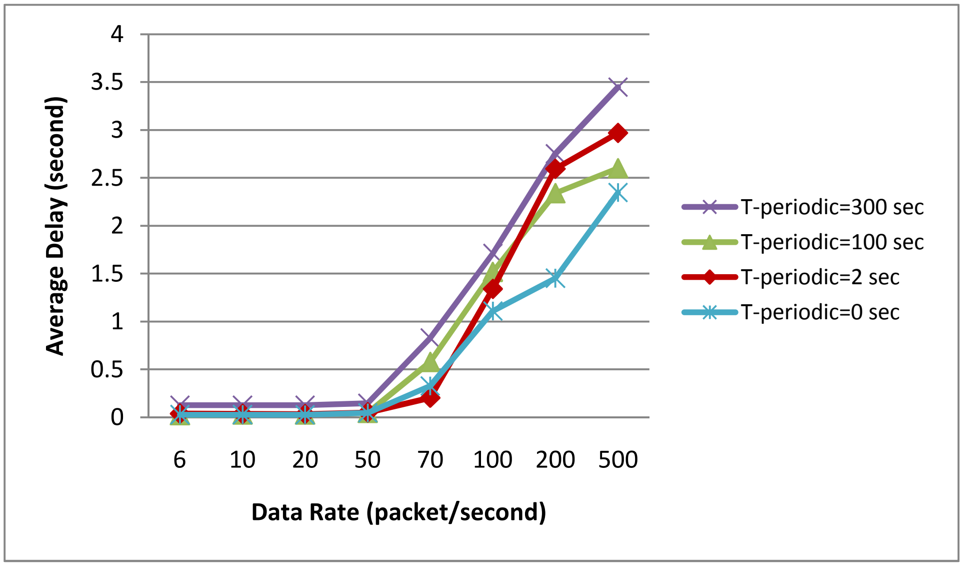


Figure 5.17: Average delay of timer swap attack with changing data rate

Comparison between Counter and Timer Swap Attack in Term of Sink Throughput

At this point, our experiment is designed to evaluate and explore the difference between counter and timer attacks. As shown in Figure 5.18, always there is a type that clearly outperforms the other. The performances of both types vary when varying the data rate and the period of the attack (for timer attack). For larger data rate, counter swap outperforms the timer attack which has the same performance for different data rate. However, for small data rate, the timer swap with large $T_{periodic}$ outperforms the counter swap which has the same performance of small $T_{periodic.}$

These results are consistent with network behavior, since for small data rate, the number of received data and corresponding control signaling is small. And as counter swap is activated upon receiving a new packets, the attack is activated for shorter time than in the timer case with $T_{periodic. =}$ 2. Nevertheless, as data rate increases, the traffic packets received by sensor nodes (data/control) increase throughout of the network and thus the attack is activated for longer time by counter attack, while in the timer attack it is limited to the attack period.

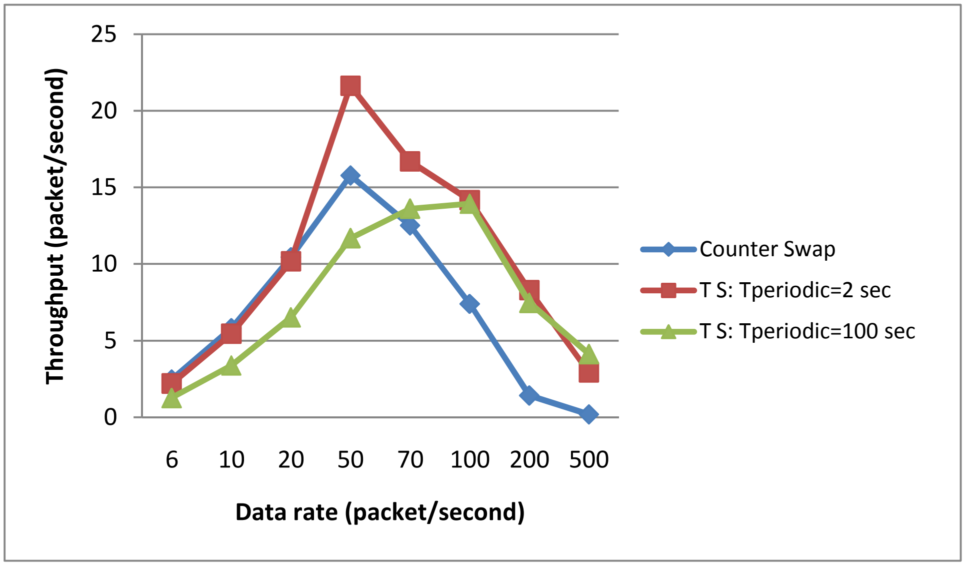


Figure 5.18: Comparison between counter and timer swap attack in term of sink throughput

The previous discussion is also convincing for Figure 5.19 which compares the two schemes in terms of average delay. We notice that at high data rate the counter mechanism exhibits better value of the delay (larger) as the number of control signaling associated with high data rate and activated upon receiving a new packet causes the nodes to consume its time in processing the incoming packets causing relatively high delay. Note that, at high data rate the period attack has no remarkable effect on both throughput and average delay as discussed previously in this context.

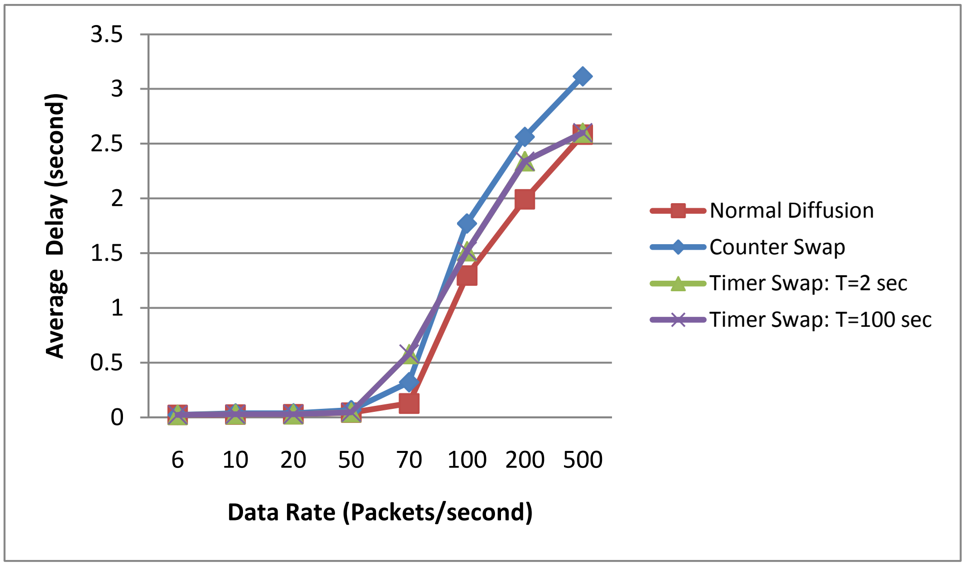


Figure 6.19: Comparison between counter and timer swap attack in term of average delay

5.5.3 Simulation Results of Bee Swarm Flooding Attack

This subsection is dedicated to report on the quantitative simulations we have performed to analyze the proposed swarm attacks in chapter 4.

Performance of Bee Swarm Attack over Time:

The system performance has been observed in four scenarios. The first scenario is that there are no attacking nodes in sensor networks. In order to carefully observe the impact of our swarm attack on performance of sensory networks, we assume that rates of attacking packets are 50 packets/s, 100 packets/s, and 150 packets/s. In other words, the attack process is launched by floods of 50, 100, and 150 packets every second. We calculate the throughput every 100s. At 100s of simulation experiment, we totalize throughput from 0 to 100s. At 200s of simulation experiment, we totalize throughput from 100 to 200s. The rest may be deduced by analogy. The results are as follows. In Figure 5.20, we observe that throughout goes down when an intruder starts to flood the attacking packets. The average throughput is 9.09 without attack and large numbers of packets get to the destination nodes. However, the throughput declines from 9.09 when the intruder floods 40 packets every second. In other words, most packets can't get to the goal and those packets are discarded by nodes for network congestion. Interestingly, the network seems to have some recoverability.

When the rate of attacking packets is less than 50 packets/s, the performance becomes better after a while. But when the rate of attacking packets is more than 150 packets/s, the network can not bear the attack anymore and the performance goes down quickly.

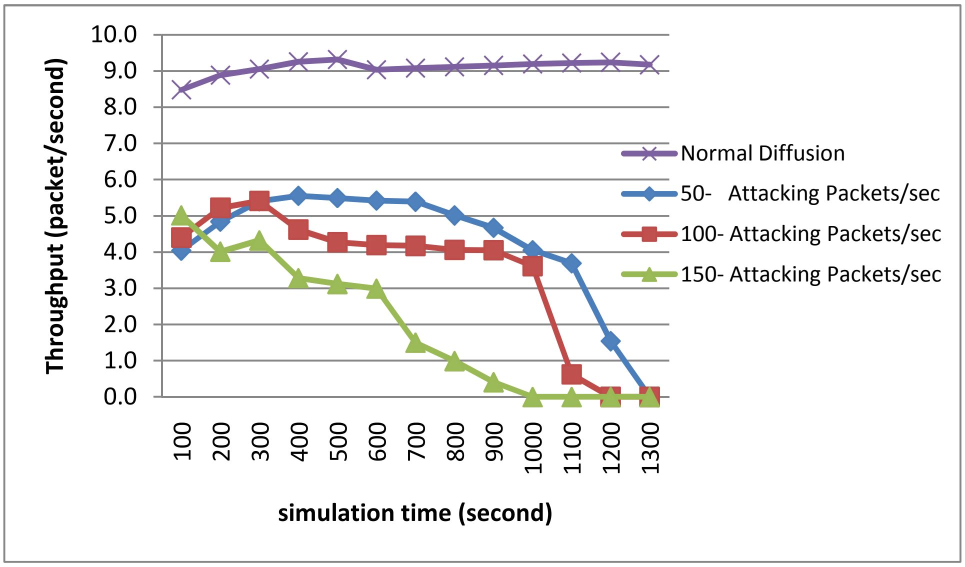


Figure 5.20: Effect of different attacking packet number on sink throughput over time

Observe that, in the previous graph, we plot the throughput for different number of attacking packets per second. However, the attacking packets depend on three factors: number of attackers, number of interests generated by each attacker, and the rate of these interests. By multiplying these three variables we could change the rate of attacking packets. Table 5.2 demonstrates that the system performance significantly varies for the same number of attacking packets with different combinations of the three factors. To explain what contributes to the throughput decline depicted in Figure 5.20, we now describe a set of six separate experiments to explore the optimum traffic pattern that the attacker can use to effectively achieve its goals. In these experiments, we study the relationship between these factors by changing one factor at a time with fixing the other two variables. By doing this, we identify the conditions in which we could accurately approximate the optimal DoS traffic pattern.

Table 5.2: System performance over different combinations of the attacking packets

Attacking Packets	Attackers	Interests	Interest Rate	T-$_{Deny}$	PDR	Throughput
250	5	1	50	1123	88.31%	3.86
	1	5	50	1122	65.79%	3.84
	8	30	1	923	51.23%	2.93

Performance of Different Number of Interest when Changing Number of Attackers

Figure 5.21 demonstrates the difference in throughput when interest rate is constant and number of attackers is variable for different fixed values of interests. The figure shows that throughput has a limited decline as increasing the number of attackers for fixed value of interest. Since DD attempts to minimize routing traffic and limits the number of identical broadcasted interests, it was designed to discard the received interest if it has a match with one of the stored interests in its cache entry. The matching between the incoming interest and those in the cache is determined by comparing their *type* and or their *rect* (region). And as interest entries in the cache don't contain information about the sink, here our attacker, but just information about the intermediately previous hop, it makes no difference if there are 2 or 20 attackers in the network as long as they produce the same data type of interest. This fact makes the limited advantage of increasing the number of attackers, represented by the partial decline in the throughput, lies in the feasibility to reach more nodes in the network not to flood more interests. However, we notice that the number of interests has a noticeable effect on degrading system performance. This result reflects the fact that as new interest is injected to the network, all the intermediate nodes should propagate this interest until it times out which would cause high traffic in the network and exhaust the resources of the network. Also, we measure the performance of our attack in terms of T_{deny}, Figure 5.22.

The results indicate that the time needed to deny the service is constant for the same number of attackers. Even for different number of interests, T_{deny} has a slight decline.

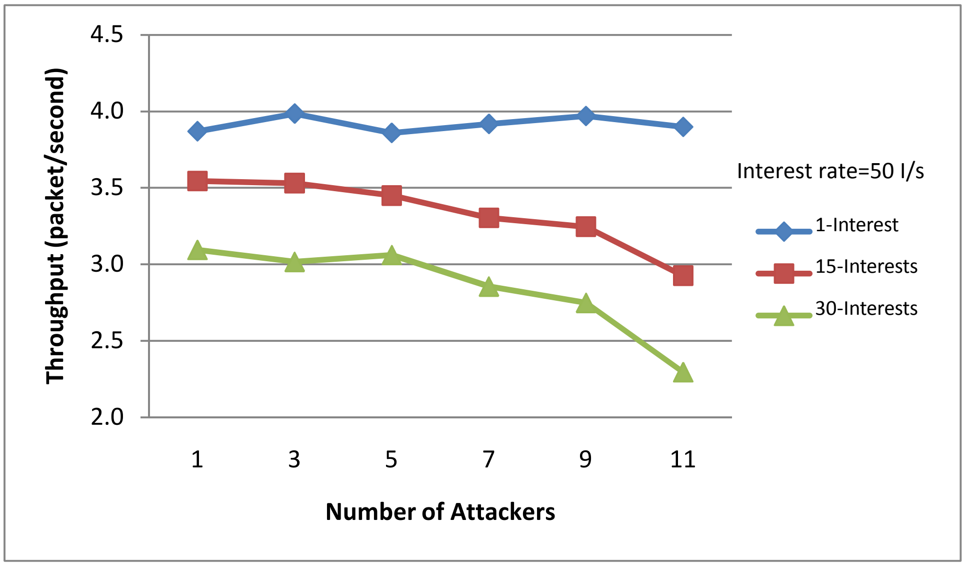


Figure 5.21: Throughput of different number of interest when changing number of attackers

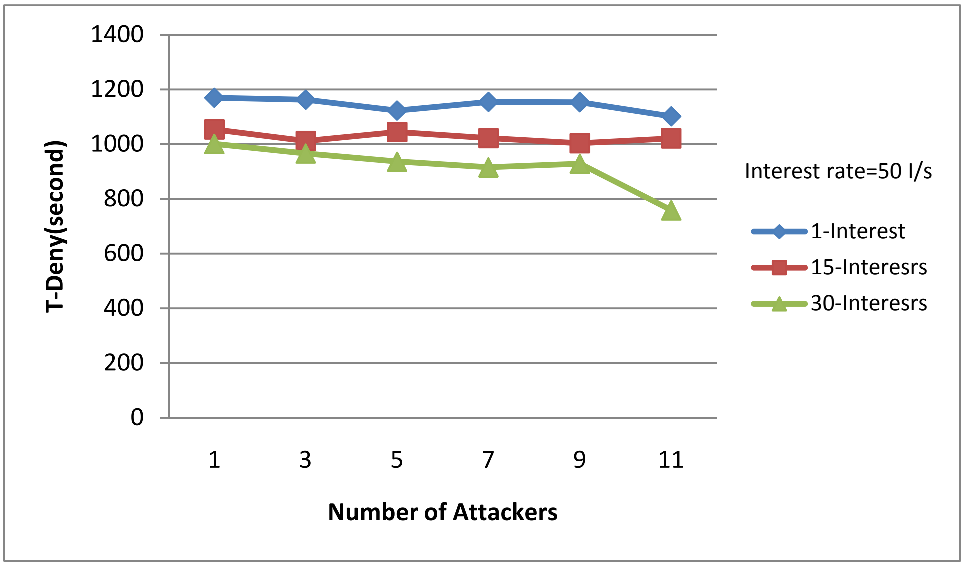


Figure 6.22: Deny time of different number of interest when changing number of attackers

Performance of Different Interest Rate when Changing Number of Attackers

Next, we evaluate the throughput and T_{deny} when the number of interests is constant by changing the number of attackers for different values of interest rate. Figure 5.23 confirms that as the number of attackers increases, the throughput decreases. However, the figure also indicates that varying the interest rate generated by each attacker has no visible effect on the performance. In other words, an attacker who diffuses an interest of the same data type with rate of 1000 interests per second has nearly the same effect if it just diffuses it with 10 interests per second. This result can be explained by DD specifications. In DD interest propagation stage, every node receives a new interest, checks to see whether this interest exists in its cache. If a similar entry exists, it simply drops the interest. However, for large values of data rate, the throughput is rapidly decreased to approximately 2.5 as the number of injected packets is very high represented by 11x100x15 which wastes the resources of legitimate sensors in processing the incoming packets. Although identical interests are eliminated, most of the intermediate nodes are busy receiving and handling the incoming fake interests.

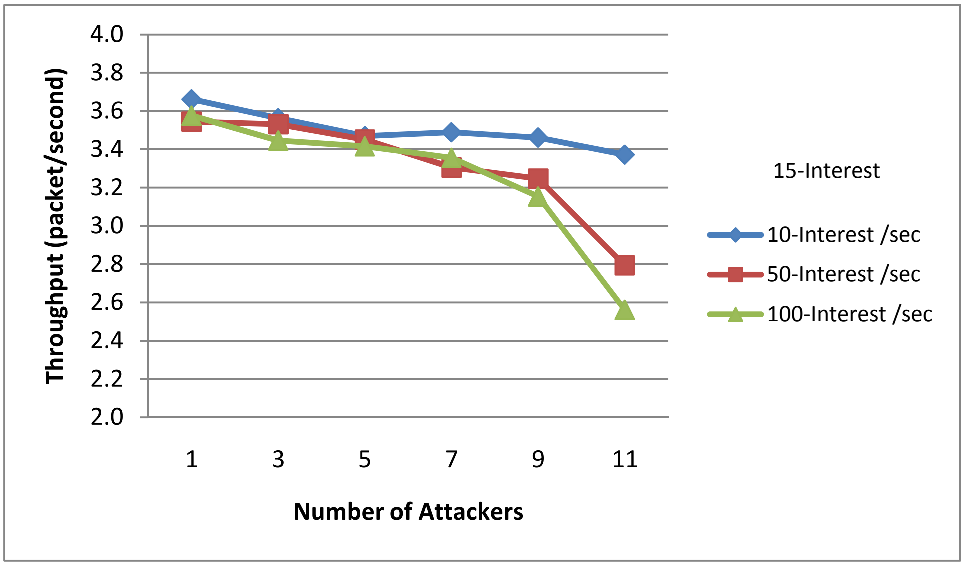


Figure 5.23: Throughput of different interest rate when changing number of attackers

Figure 5.24 proves that for relatively large number of attackers, the service would be denied more quickly for larger rate of interest as the previous discussion reveals.

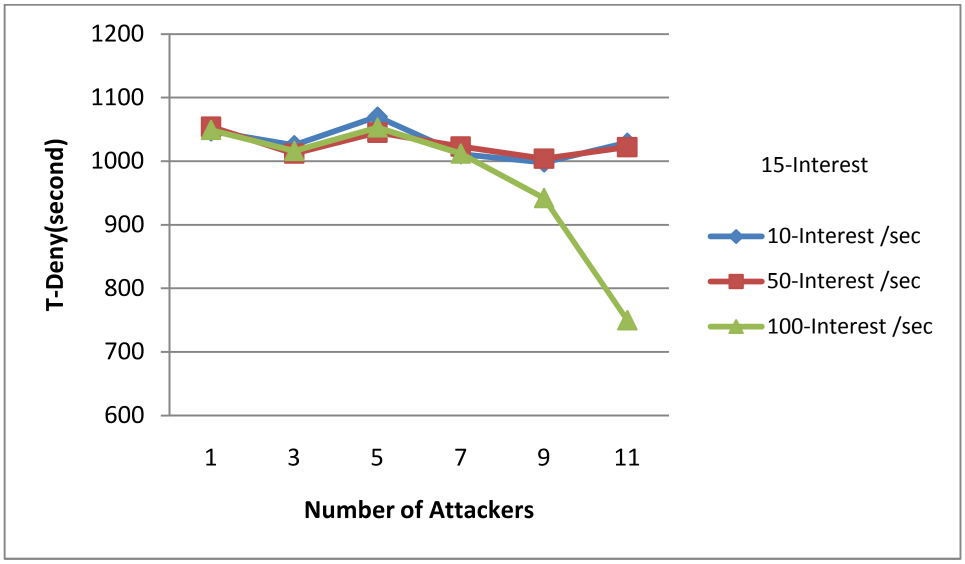


Figure 5.24: Deny time of different interest rate when changing number of attackers

Performance of Different Attacker's Number when Changing Number of Interests

Figures 5.25 and 5.26 confirm the result obtained in the previous experiments. As mentioned earlier, this result implies that the number of interests is the dominant aspect that can influence the throughput of the sink. While number of attackers has partial effect, the interest rate is unable to produce any significant improvement over our attack scheme.

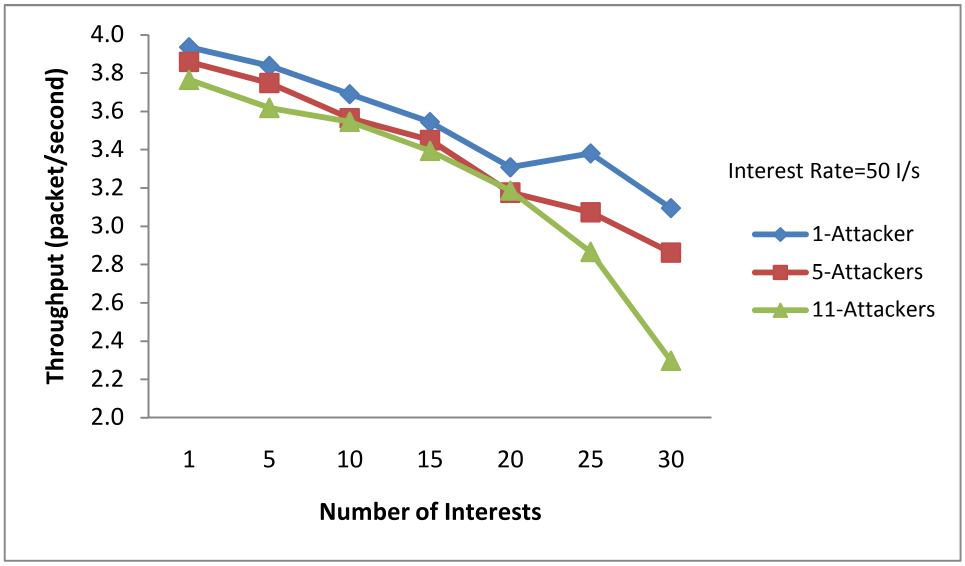


Figure 5.25: Throughput of different attackers' number when changing number of interests

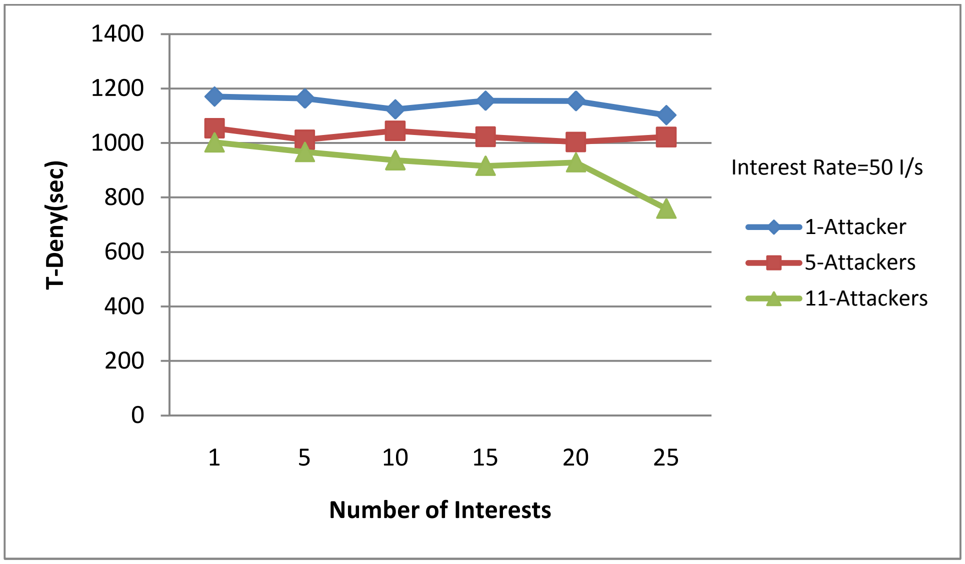


Figure 5.26: Deny time of different attackers' number when changing number of interests

For deny time, the effect is remarkable for the larger resultant number of attackers and number of interests where relatively sharp decline in deny time is observed in 11-attackers/25-interests scenario.

<u>Performance of Different Interest Rate when Changing Number of Interests</u>

Figure 5.27 proves that the number of interests is the main factor that influences the throughput regardless the value of the data rate.

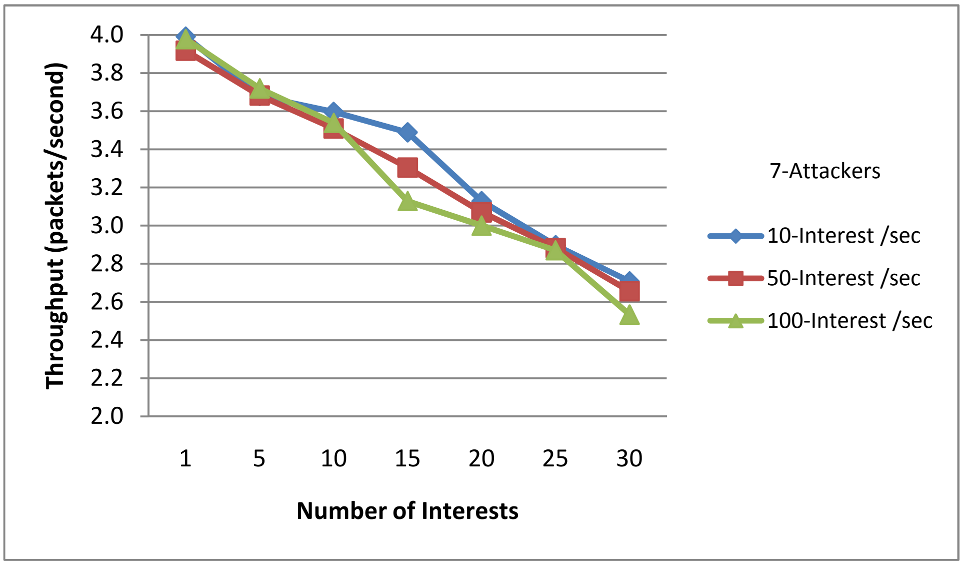


Figure 5.27: Throughput of different interest rate when changing number of interests

In addition, and as Figure 5.28 indicates, T_{deny} slightly decreases with changing interests' number. However, the interest rate has absolutely zero effect on deny time of the system except for high rates (100) as it consumes the sensor times in processing incoming packets.

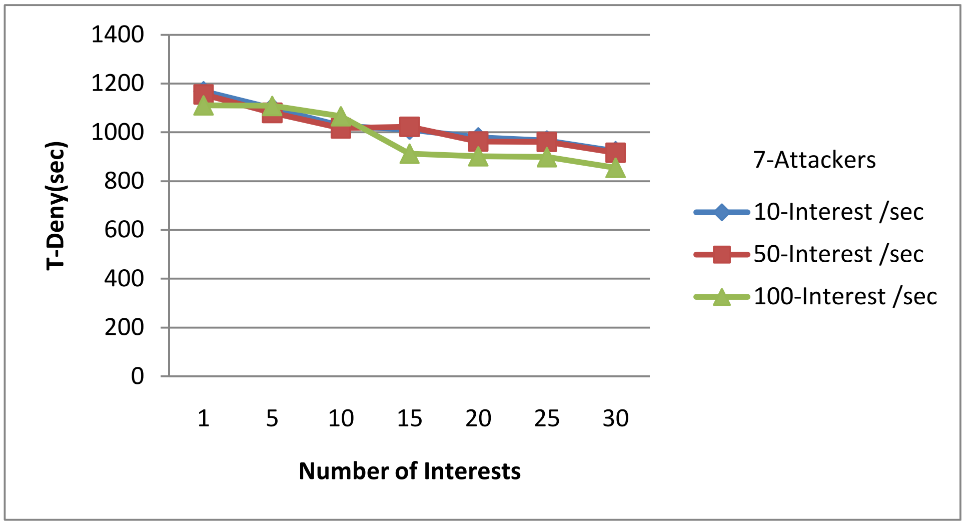


Figure 5.28: Deny time of different interest rate when changing number of interests

Performance of Different Attacker's Number when Changing Interest Rate

Although we have shown that interest rate doesn't affect the network behavior, more investigation would show some effect. These effects are not visible in previous figures. It is easily observed from Figure 5.29 that the behavior of the three curves is close.

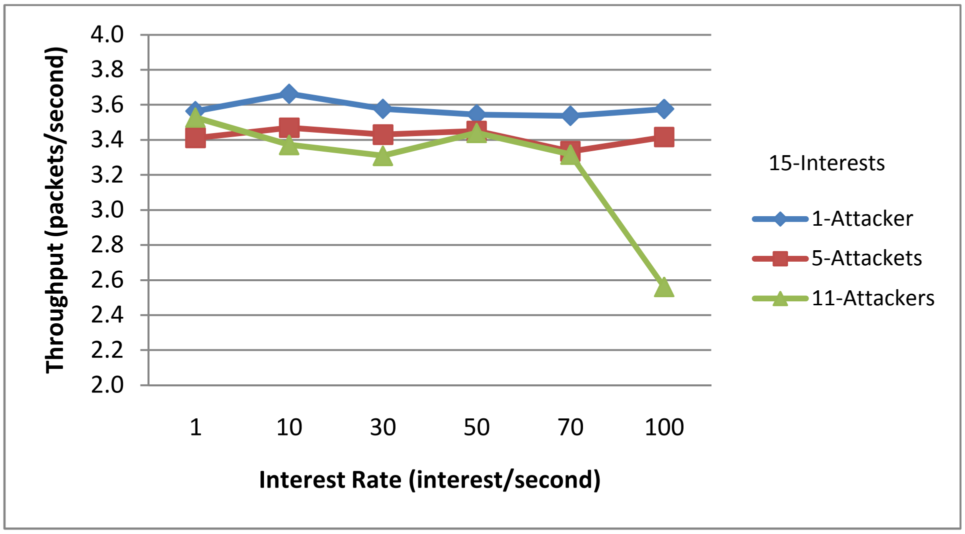


Figure 5.29: Throughput of different attackers' number when changing interest rate.

However, the effect of our attack is more prominent when both data rate and the number of attacker are at their maximum. This justification is also valid for T_{deny} in Figure 5.30.

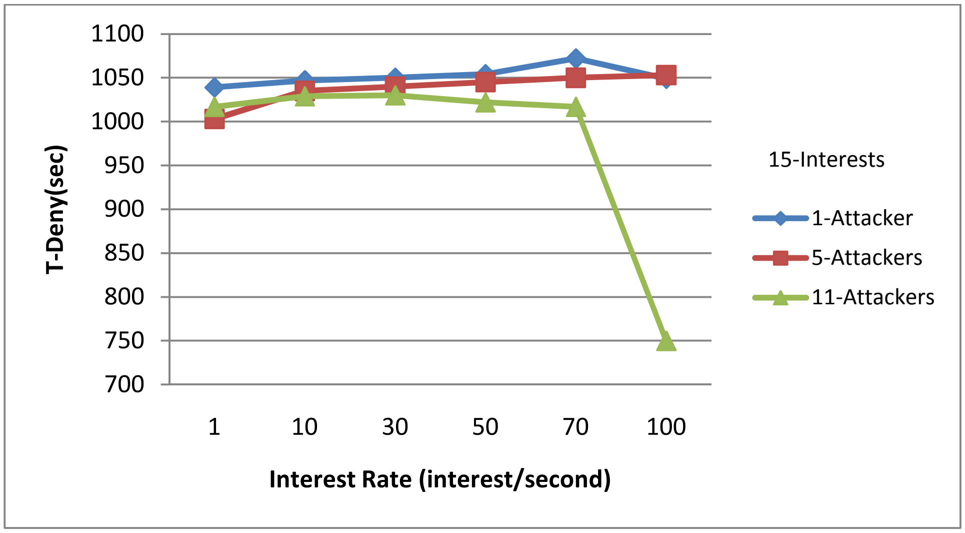


Figure 5.30: Deny time of different attackers' number when changing interest rate.

Performance of Different Interest's Number when Changing Interest Rate

Again, the last experiment of this series to explore the space parameters of attacking packets rating confirms the previously obtained results and summarizes the result in Figure 5.31 and Figure 5.32. Number of interests is the dominant factor, and for high products of the three variables, a significant degradation is obtained in relatively small deny time.

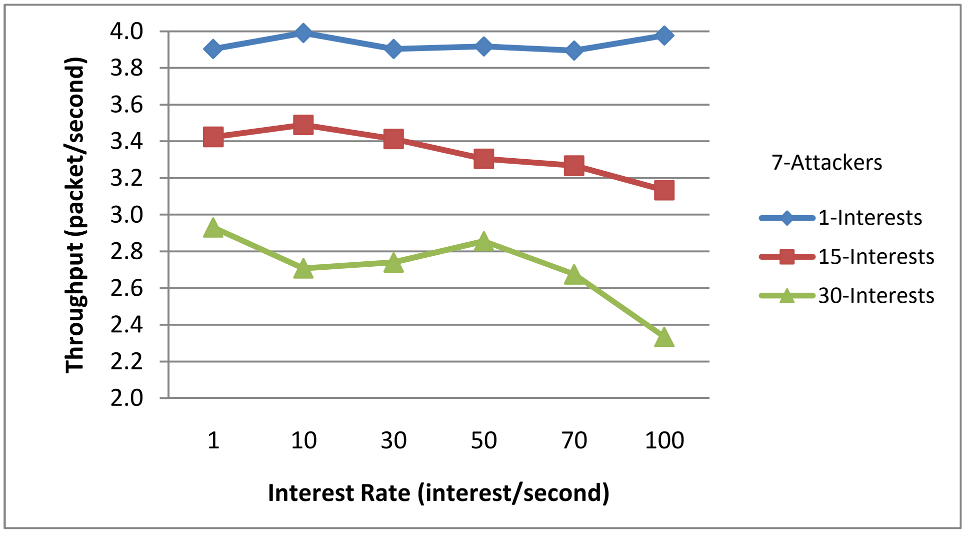


Figure 5.31: Throughput of different interests' number when changing interest rate

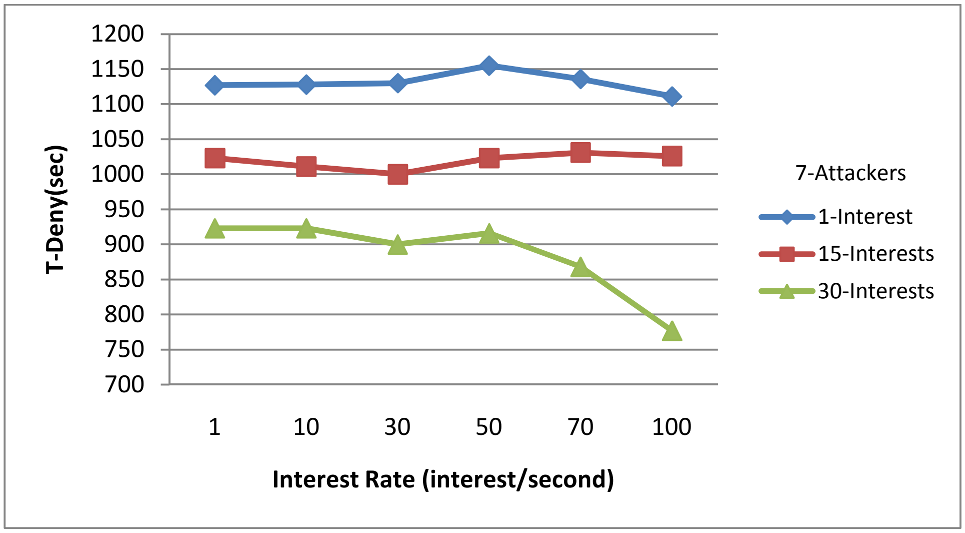


Figure 5.32: Deny time of different interests' number when changing interest rate

5.5.4 Simulation Results of Ant Swarm Flooding Attack

The previous simulations illustrate how Bee Swarm Attack can severely degrade the throughput of the network. Here, we investigate how to produce another efficient attack named Ant Swarm Attack and see how we can utilize Ant Swarm Attack to obtain different and more efficient performance of Bee Swarm Attack by changing the attack parameters. The results obtained previously on Bee Attack also apply here as Bee Attack is a special case of Ant attack with T-delay = 0. We ran two set of simulations where 3, 5, 7, and 10 attackers are injected to the network on bursts with T-delay is the timing separation between these bursts; the first experiment represents sequential and individual entry of attackers while the second one describes the entry of attackers on bursts in hierarchical manner of timing entry.

Performance of Sequential Ant Swarm Attack

Our simulations consist of a variety of network configurations and traffic patterns simulating both sequential as well as hierarchical attacks coming from multiple and variable distributed attackers. For simulating attacks from different attackers, we use different delay values for the entry of the attacker. The collected statistics are used to plot throughput against attack inter-burst period. The throughput value provides the metric for evaluating the efficiency of our algorithm, and for comparing the results with bee attack.

We carried out an experiment to show the effect of sequential attack on network behavior. Figure 5.33 reveals sequential behavior of our attack. As it can be seen, for small values of difference between the entry of attackers, as the number of attackers increases, the throughput decreases and our attack is more successful. While increasing the delay causes the order of curves to be reversed and the smallest number of attackers gives the more efficient attack. This can be explained by the fact that for small delays, all the attackers enter the network sequentially with negligible delay, which means that for delay equals 3, for example, after 9, 15, 21, and 30 seconds all 3, 5, 7, and 10 attackers would be in the network. However, as the delay increases, more time is needed for the larger number of attackers to enter the network and participate in the attack. For example if T_{delay} is 200, in the case of 10 attackers only at time of 1000 second all the 10 attackers were available in the network. However, the 3 attackers would be completely effective at 300 second. Note that, the difference between the performances as sequentially attacking the network is mostly dedicated to the sequential entrance of interests and not the attackers themselves. This is because unlike the bee attack in which the attackers flood similar interests, in ant attack the attackers flood sequential interests, i.e. for 3 attackers of T_{delay} 10, the attackers enter the network at times 0, 10, and 20 with interest of data type (0-9)(10-19)(20-29) for the three attackers individually.

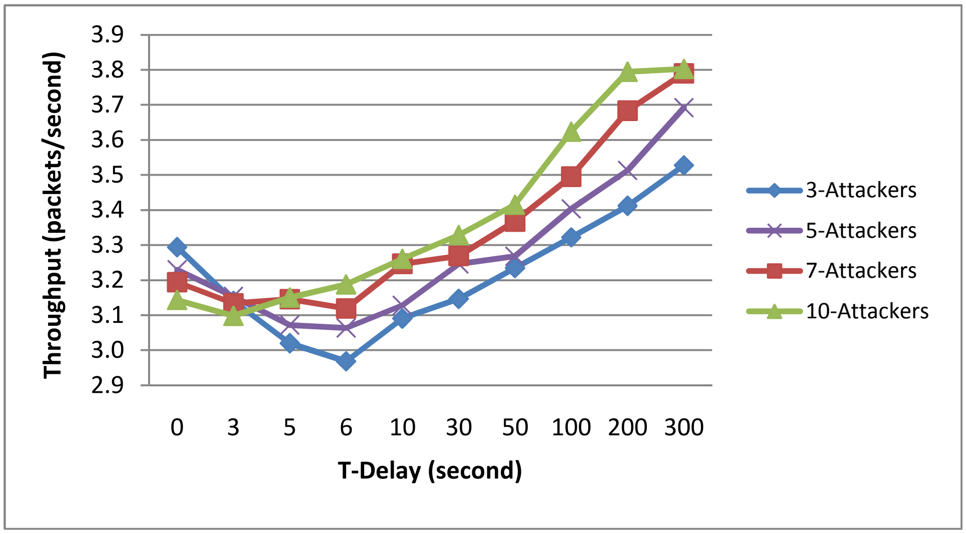


Figure 5.33: Performance of ant sequential attack in term of sink throughput

Performance of Hierarchical Ant Swarm Attack

We further investigate two types of this attack; the first one is top-to-base hierarchical in which the attackers enter the network in hierarchal pattern starting with a small burst followed by gradually increasing other bursts. The second type is base-to-top hierarchical in which the bursts of the hierarchical attack have been reversed. For our experiment of 3, 5, 7, and 10 attackers, Table 5.3 demonstrates the bursts of both types of attack.

Table 5.3: Illustration of bursts of forward/reverse hierarchical attack

Attackers' Number	3	5	7	10
Top-to-Base (Forward)	1-2	2-3	1-2-4	1-2-3-4
Base-to-Top (Reverse)	2-1	3-2	4-2-1	4-3-2-1

The performances of these attacks are plotted in Figure 5.34 and Figure 5.35. At first glance, one may think that three flooding ant schemes can provide similar behavior. Further inspection, however, reveals the difference. It is depicted that the curves behavior swap at earlier time in Figure 5.35 compared to Figure 5.34 and earlier in Figure 5.34 compared to Figure 5.33. This is due to the artifact that as more attackers are in the network earlier, more different interests are flooded to the network and the effect of the attack appears faster.

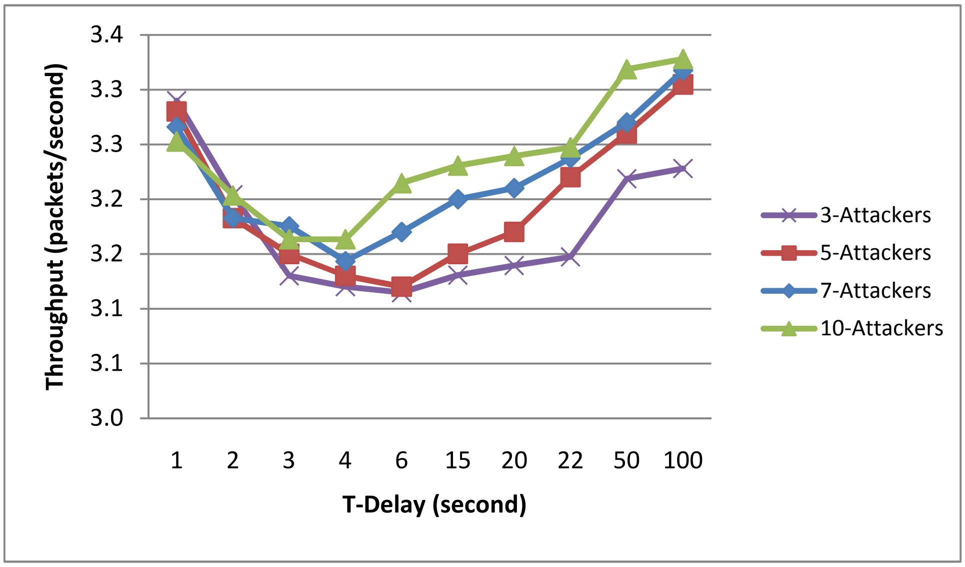


Figure 5.34: Performance of top-to-base hierarchical ant attack in term of sink throughput

Comparing the three schemes, the curves of the different attackers have been swapped at 6, 4, and 2 seconds for sequential, hierarchical and reverse hierarchical, respectively. Also, note that the amplitude of the throughput decreases in reverse hierarchical compared to the hierarchical.

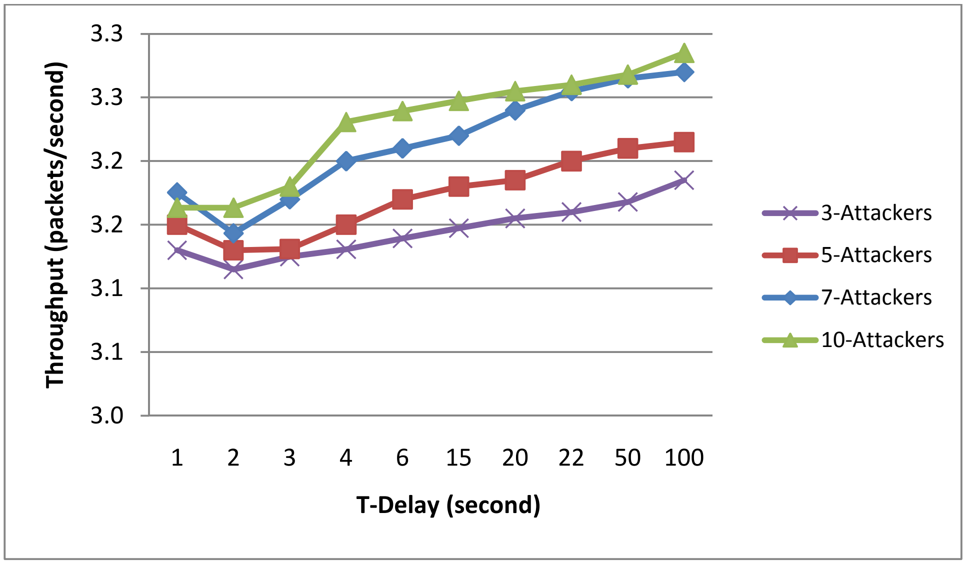


Figure 5.35: Performance of base-to-top hierarchical ant attack in term of sink throughput

Comparison of Different Swarm Attack

Throughout this chapter, we have discussed distinct schemes of Swarm attacks. We conducted another simulation to compare among these schemes. Figure 5.36 and Figure 5.37 measure the attack capabilities of four schemes aiming to degrade the throughput at sink node. For each scheme, we fix the interest rate while changing the number of attackers. The figures show that Bee Swarm Attack with 30 interests per attacker is superior to the other schemes as it causes the maximum decrease in throughput and denies the service earlier. Note that for Ant Attack, we consider 30 interests in the whole network divided equally by the specified number of attackers. At the first glance, it may seem that bee swarm attack is superior to other schemes. However, further inspection reveals that Ant Swarm is competitive despite the fact that bee could achieve more degradation in sink throughput.

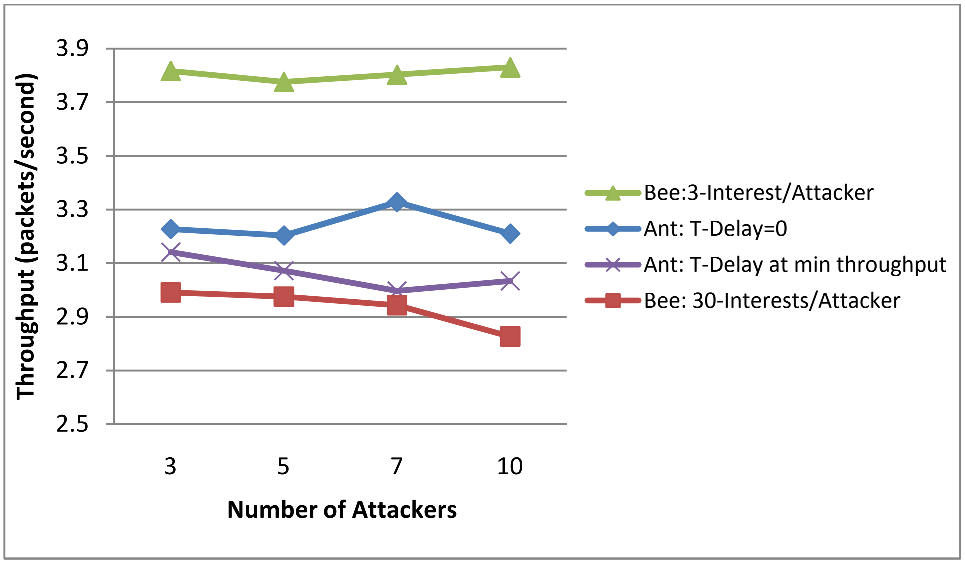


Figure 5.36: Comparison of different swarm attacks in term of sink throughput

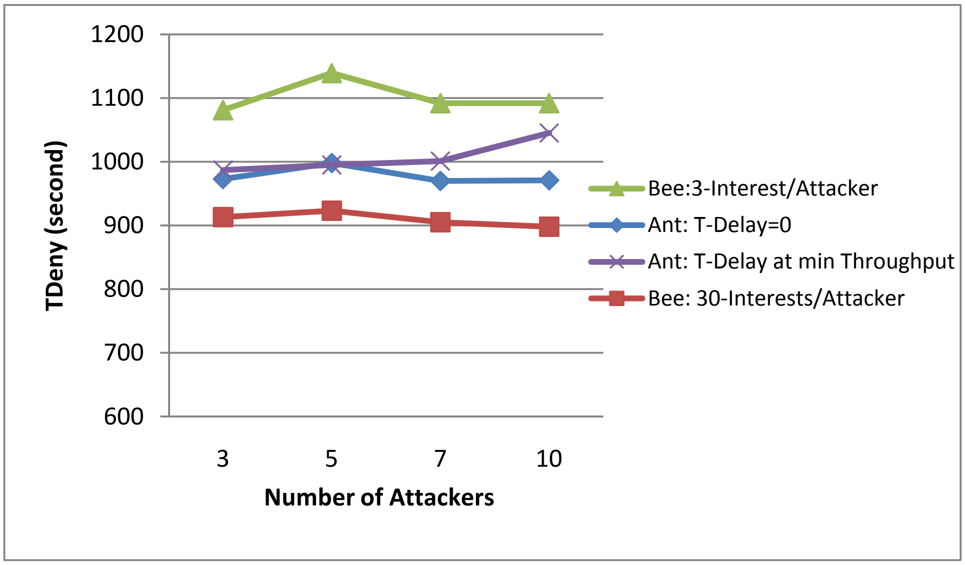


Figure 5.37: Comparison of different swarm attacks in term of sink deny time

For Bee Swarm attack, every attacker has to flood 30 interest types while in the Ant Swarm Attack, the maximum allowable interest type (which is 30) is divided equally between available attackers. For Figure 5.36, every one of the three attackers only floods 10 interests, which means conservation in attacker resources. Even for Ant with delay equals zero, it gives comparable results. The same behavior has been obtained when comparing the two swarm approaches in terms of average delay. The results are presented in two separate figures for scaling issues. Both figures (Figure 5.38 and Figure 5.39) show that bee outperforms ant in increasing the packet delivery delay noting the difference in interest number disseminated by each attacker in both cases.

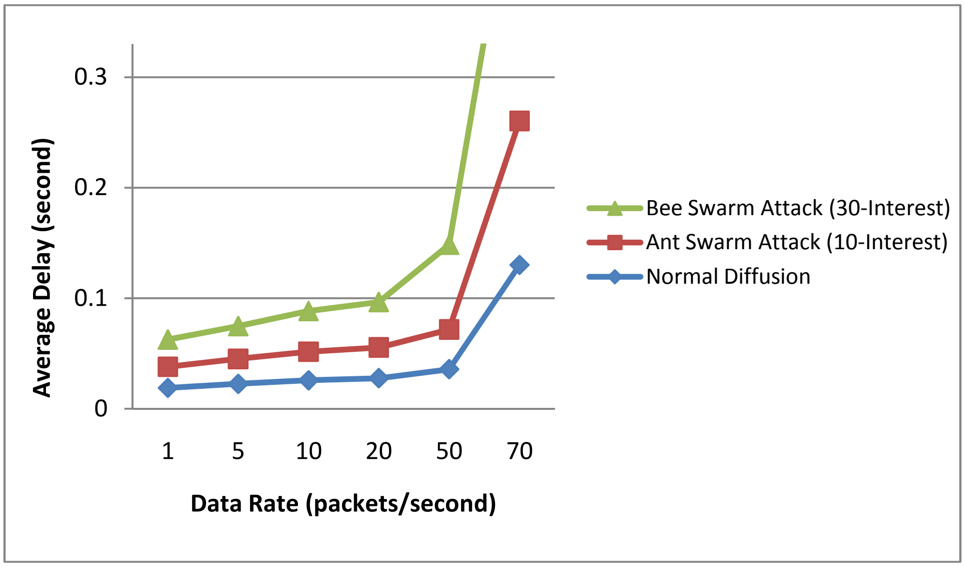


Figure 6.38: Comparison of different swarm attacks in term of average delay with small data rate

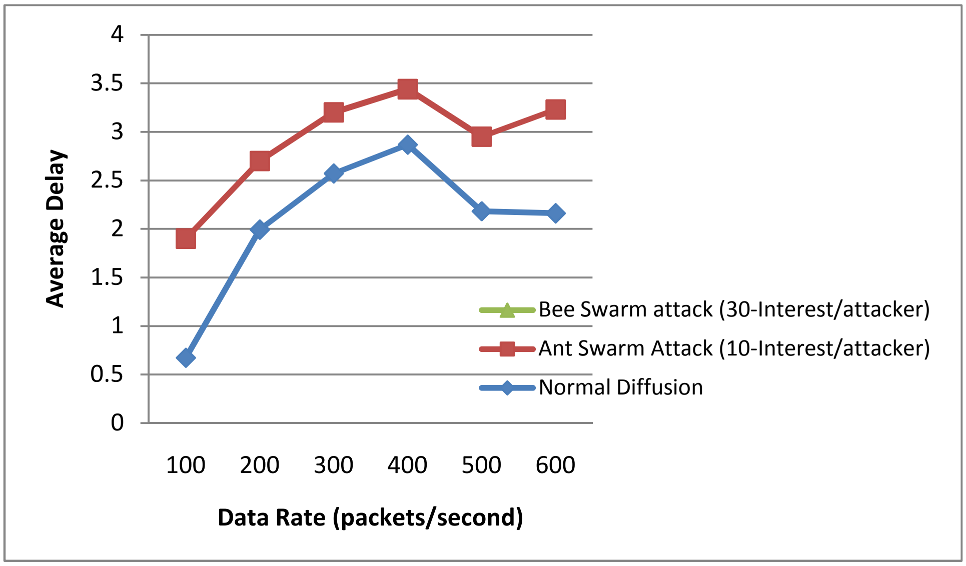


Figure 6.39: Comparison of different swarm attacks in term of average delay with high data rate

Performance of Sequential Ant Attack over Multiple Sinks Network

Next we consider the multiple-sink scenario. The experiment is repeated with increasing number of sinks up to 7 sinks so as to find out the impact of attack streams if the attacks are launched against multiple sinks network. This kind of scenario is one of the most important cases to judge the success of the attack as the attacker would be able to deny the service for multiple sinks distributed across the network. The effect is seen in Figure 5.40 as the victim network is similar to the normal diffusion but with less throughput values. Figure 5.41 also indicates that our attack decreases the number of data sent by source.

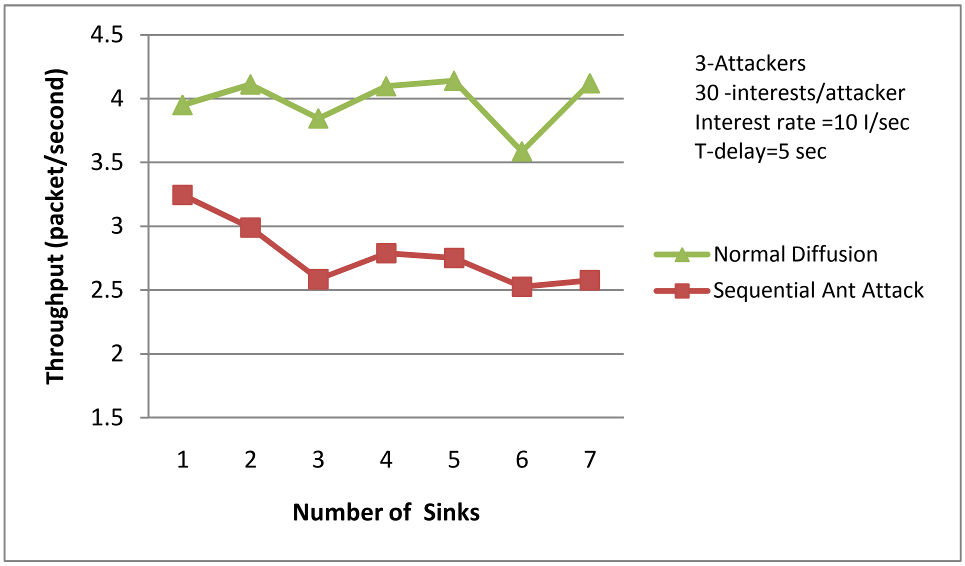


Figure 5.40: Performance of sequential ant attack over multiple sinks network

We plot T_{deny} as a function of number of sinks in Figure 5.42. For this experiment, we estimate the value of the system deny time by taking the maximum value of T_{deny} obtained for the specified number of the sink nodes in the system.

We depict an increase in T_{deny} as increasing system sinks as more time is needed to saturate multiple links in which the data is transferred from source to sinks.

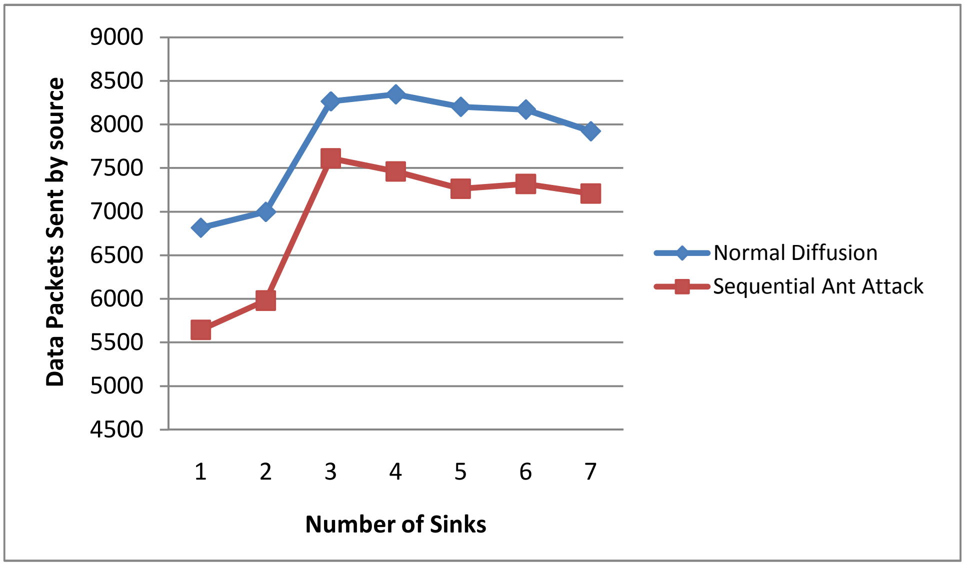


Figure 5.41: Number of data packets sent by source in multiple sinks network

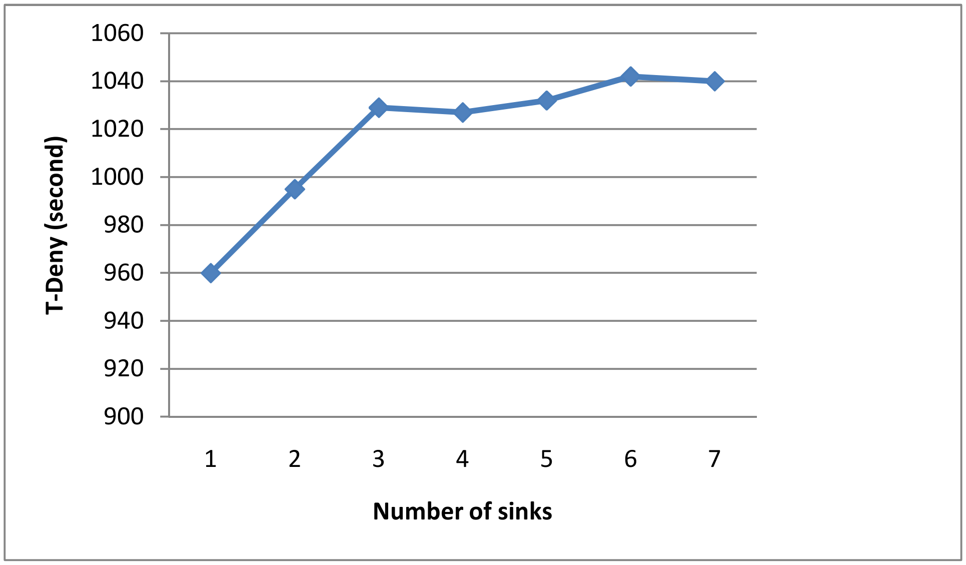


Figure 5.42: Deny time under sequential ant attack in multiple sinks network

5.6 Attack Strength of the Simulated Attacks

In this chapter, we present a wide range of experiments to simulate multiple techniques of attacks; the results obtained in our simulations indicate that all the proposed attacks can significantly degrade the network performance either by decreasing the throughput or increasing the system delay. While each of the simulated attacks can cause substantial destruction to DD routing protocol, we further prefer to compare between these different schemes of attacks.

In [59], the authors defined relative strength of a particular attack configuration Σ, which represents the amount of damage an attack can cause per adversary, as:

$$\Sigma = \frac{DR_{norm} - DR_{adv}}{DR_{norm} \, . \, Num_{adv}} \tag{5.1}$$

Where DR_{norm} and DR_{adv} are the delivery ratios in the absence or in the presence of the attacker respectively, and Num_{adv} is the number of attackers. We adapt the previous formula in terms of throughput and apply the modified formula to all the proposed attacks and report the results in Table 5.4.

Table 5.4: Attack strength of the simulated attacks.

Attack Type	Attack Strength
Counter Swap Attack (Halt Mode)	24.44
Counter Swap Attack (Norm Mode)	10.59
Timer Swap Attack (Norm Mode)	12.58
Bee Swarm Attack	13.98
Sequential Ant Swarm Attack	12.38
Hierarchical Ant Swarm Attack	12.53
R- Hierarchical Ant Swarm Attack	12.59

The results indicate relatively high attack strength compared to values obtained in [59]; for their attacks, they obtained the value of 23.4 as the highest observed attack strength out of all considered attacks. While they have the most values close to 13.

Chapter 6

CONCLUSION

This thesis has shown, through modeling and implementation, the susceptibility of modern WSN routing protocols to devastating denial-of-service attacks. A detailed analysis of denial-of-service vulnerabilities of WSN particularly Directed Diffusion protocol, along with a description of attacks that target these vulnerabilities, makes evident the ease with which attacks can be launched against this protocol. Encrypting and authenticating network traffic is not sufficient to protect networks from denial-of-service attacks.

In this research, we have conducted an intensive and detailed study on DoS attack in WSNs with the main focus on Directed Diffusion protocol and its vulnerabilities.

Throughout this thesis, we have introduced two new attacks against DD based WSN, namely Reinforcement Swap Attack and Swarm Flooding Attack.

Swap Attack, our first attack model is based on disturbing route discovery phase in DD operation. This approach has been done via swapping reinforcement rules of the protocol. In the original operation of DD, it is stated that short delay route is elected using positive reinforcement while the high delay route is eliminated using negative reinforcement. However, in our attacked version of DD we swap these rules to include the bad paths and exclude the good paths. Our swap attack has been implemented in more than one approach. The first approach we present is counter swap attack which aims to alternate between the original and the swapped rule of reinforcement on receiving a new packet. Timer swap attack is also proposed as an alternative to counter except that the attack switches between the on and off period of the attack based on previously determined time slot of attack period.

Our analysis points out several key features of Swap Attack. We found that counter attack performs well in some applications but poorly in others. In high data rate application (like surveillance of valuable things which need continuous feedback of the current status of the system), counter attack performs better than timer attack. However, for small data rate timer attack with moderate to large periodic attack outperforms counter attack. In addition to these two mechanisms of attack implementation, our swap attack can be activated on two modes: Halt mode which results in fast and fatal disruption of the network and Norm Mode which allows the attacker to insert itself in the network, gradually affect the network, and finally degrade performance but with more than the time needed by Halt mode. The aim of the attacker is the key factor which identifies which modes to activate and whether the attacker is interested in rapid interruption in sensor communication or it rather prefers to be able to participate in the network operation as long as possible.

In addition to Reinforcement Swap Attack, we proposed another new attack which allows an attacker to mount DoS attack against most of currently proposed on-demand routing protocols. This attack integrates the concepts of swarming and flooding. Swarm Flooding Attack is based on the idea of attacking the victim network with multiple well coordinated swarms of attackers. The attacking process is accomplished via flooding the system with excessive number of packets.

We prove that attacking a target from many locations could be done in different ways. Bee Swarm Attack is the first model which we validate through our simulation using NS-2 simulator. Bee attack is simple, easy to launch, however, it requires the synchronization between sensors in order to launch the attack. We explore the parameter space of bee attack and it is found that swarm number or capacity in terms of attackers' number is not the dominant here. The significant factor here is the swarm capacity in terms of number of injected interests into the network. These results strengthen our attack in a way that it could be done efficiently by single powerful well positioned attacker. The results indicate that increasing number of attackers has slight effect on the success of the attack.

A second way of swarm attack is to follow ant swarming technique. Ant swarm is different from Bee swarm in which they move in linear formations, but can shift into swarming mode when it is time to attack. We simulate three formations of Ant Attack; all of them give remarkable decline in network throughput and increase in average delay. However, Bee Swarm Attack outperforms Ant Swarm Attack but noting that in Bee Swam, every attacker has to produce exactly the same interests as other attackers even if identical interests are suppressed by intermediate nodes. On the other hand, although ant attack reduces the throughput in a less rate than bee, the attack is more efficient since it conserves the resources of attacking sensors. For classification of both ant and bee attacks, both of them need synchronization devices between attackers which may consume the resource of the attacker, so Ant Swarm Attack is more suitable for limited capabilities attacker while Bee Swarm Attack could be classified as lap-top class attack.

Also, we analyzed the relative strength of the attack in terms of the magnitude of disruption caused per attacker, and all of our attack could achieve relatively strong values in this context compared to standard and well known routing attacks.

This work, which compares a number of distinct attacking models, would provide additional insights. Specifically, it would draw conclusions regarding the choice of the best suited protocol to be employed in a precisely predefined realistic application.

This research re-emphasizes the importance of considering security early in the network protocol development process. Without this, vulnerabilities inherent in these network protocols, and other software, will increasingly become targets for malicious attacks.

REFERENCRES

[1] A. D.Wood, J. A Stankovic,‘Denial of Service in Sensor Networks’, *IEEE Computer*., vol. 35, no. 10, October 2002, pp. 54-62.

[2] C. Intanagonwiwat, R. Govindan and D. Estrin, "Directed diffusion: A scalable and robust communication paradigm for sensor networks", in *Proc. 6th Annual ACM/IEEE MobiCom'00*, Boston, MA, August 2000.

[3] K. Fall and K. Varadhan,‘Editors ns Notes and Documentation,’ The VINT Project, UC Berkeley, LBL, USC/ISI, and Xerox PARC, Nov. 1997. Available: http://www-mash.cs.berkeley.edu/ns

[4] P Pancardo, JC Dueñas, “A proposal for System Architecture to Integrate Scarce-resources Wireless Sensor Neworks into Ubiquitous Environments”. [Online], Available: http://ftp.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-208/paper23.pdf

[5] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “Wireless sensor networks: A survey,” *Computer Networks (Elsevier)*, vol. 38, no. 4, pp. 393-422, Mar. 2002.

[6] L Wang, J May Gao, SJ Moon, ‘Node-failure Tolerance of Topology in Wireless Sensor Networks’, [Online], Available: http://www. ijns.nchu.edu.tw

[7] C. Y. Chong, Kumar, SP,‘Sensor networks: evolution, opportunities, and challenges’, in *Proc. of the IEEE*, vol. 91, no. 8, pp. 1247- 1256, Aug. 2003.

[8] P. Ning, A. Liu, and WL Du,‘Mitigating DoS attacks against broadcast authentication in wireless sensor networks, ACM Transactions on Sensor Networks, vol. 4, no. 1, pp. 35, Jan, 2008.

[9] KW Jang, SH Lee, MS Jun, ‘Design of Secure Clustering Routing Protocol using SNEP and μ TESLA on Sensor Network Communication’, *IJCSNS International Journal of Computer Science and Network Security*, vol.6, no.1B, January 2006.

[10] J. Deng, R. Han, and S. Mishra,“A Performance Evaluation of Intrusion-Tolerant Routing in Wireless Sensor Networks,” In *Proc. of the 2nd IEEE International Workshop on Information Processing in Sensor Networks*, Palo Alto, CA, USA, April 2003, pp.349–364.

[11] Z. Benenson “Authenticated Queries in Sensor Networks”, *Second EuropeanWorkshop on Security and Privacy in Ad Hoc and Sensor Networks*, Visegrad, Hungary, ESAS’05, LNCS 3813, 2005, pp. 54-67.

[12] C. Hartung, J. Balasalle, R Han, “Node compromise in sensor networks: The need for secure systems”, Univ. of Colorado, Colorado at Boulder, Tech. Report CU-CS-990-05, Jan 2005.

[13] E. Shi, and A. Perrig, “Designing secure sensor networks, Journal of IEEE Wireless Communications”, vol. 11, issue 6, Dec. 2004, pp. 38-43.

[14] Padmavathi, G., & Shanmugapriya, “A survey of attacks, security mechanisms and challenges in wireless sensor networks”. International Journal of Computer Science and Information Security (IJCSIS): vol. 4, no.1 & 2, Dec. 2009.

[15] N. Ahmed, S. S. Kanhere, S. Jha, “The Holes Problem in Wireless Sensor Networks: A Survey”, *ACM SIGMOBILE* Review, vol. 9, issue 2, April 2005.

[16] T. Dimitriou and I. Krontiris, “Autonomic communication security in sensor networks”, In I. Stavrakakis and M. Smirnov, editors, Autonomic Communication WAC, vol. 3854 of Lecture Notes in Computer Science, pp.141-152. Springer, 2005.

[17] Y. Zhou; Y. Fang; Y. Zhang, “Securing Wireless Sensor Networks: A Survey”, IEEE Communications Surveys & Tutorials”, vol.10, issue 3, pp. 6–28, 2008.

[18] Karlof, C. and Wagner, D. Secure routing in wireless sensor networks: Attacks and countermeasures. *In Proc. of the 1st IEEE International Workshop on Sensor Network Protocols and Applications*, Anchorage, AK, May 11, 2003.

[19] D. J. Thuente and M. Acharya, “Intelligent Jamming in Wireless Networks with Applications to 802.11b and Other Networks”, *in Proc. Military Communications Conf.* Atlantic City, USA, (MILCOM) 2006.

[20] A. D. Wood, J. A. Stankovic, and G. Zhou, “DEEJAM: Defeating Energy-Efficient Jamming in IEEE 802.15.4-based Wireless Networks”, *Fourth Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, San Diego, California, USA (SECON) June 18-21, 2007.

[21] S. Khattab, D. Mosse, R. Melhem, “Modeling of the Channel-Hopping Anti-Jamming Defense in Multi-Radio Wireless Networks”, *Proc. of the International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services,* Dublin, Ireland (Mobiquitous), July 2008.

[22] Z. Benenson, E. Hammerschmidt, F. Freiling, S. Lucks, L. Pimenidis, 'Tampering with Motes: Real-World Attacks on Sensor Networks', *3rd International Conference on Security in Pervasive Computing*, York, UK, (SPC) 2006, pp.38-49.

[23] Vijay Bhuse, 'Lightweight Intrusion Detection:' A Second Line of Defense for Unguarded Wireless Sensor Networks:' Ph.D. dissertation, Depart. Comp. Science. Western Michigan University, 2007.

[24] S. Chen, and Z. Zhang. "Localized algorithm for aggregate fairness in wireless sensor networks," *in Proc. of the 12th annual international conference on Mobile computing and networking*, NY, USA,(MobiCom 06) ACM 2006, pp. 274-85.

[25] T. Kavitha, D. Sridharan, (2010). "Security vulnerabilities in wireless sensor networks: a survey". *Journal of Information Assurance and Security*, vol.5, pp.31-44. Available: http://www.mirlabs.org/jias/Volume%205-Issue%201/kavitha.pdf

[26] EL Caballero, "Vulnerabilities of Intrusion Detection Systems in Mobile Ad-hoc Networks - The routing problem", 2006. Available: http://www.tml.tkk.fi/Publications/C/22/papers/Jimenez_final.pdf

[27] J.P. Walters, Z. Liang, W. Shi, and V. Chaudhary, "Wireless sensor network security: A survey". Security in distributed, grid, and pervasive computing. Auerbach Publications, CRC Press, ISBN 0-849-37921-0, 2006.

[28] Y. Wang, G. Attebury, B. Ramamurthy, "A survey of security issues in wireless sensor networks," IEEE Communications Surveys and Tutorials, vol. 8, no. 2, pp. 2-23, 2006.

[29] T. Martin, M. Hsiao, D. Ha, and J. Krishnaswami, 'Denial-of-Service Attacks on Battery-powered Mobile Computers', in Second IEEE International Conference on Pervasive computing and Communications, Orlando, Florida, USA, (PerCom2004), pp. 309-318.

[30] S. Marti, T. Giuli, K. Lai and M. Baker, "Mitigating routing misbehavior in mobile ad hoc networks," *in Proc. of ACM International Conference on Mobile Computing and Networking* , Boston, Massachusetts, USA, (MOBICOM), 2000.

[31] F. Hu and N. K. Sharma, "Security considerations in ad hoc sensor networks", *Ad Hoc Networks*, vol. 3, issues 1, 2005, pp. 69-89.

[32] A. Agah and S. K. Das, 'Preventing DoS Attacks in Wireless Sensor Networks: A Repeated Game Theory Approach', International Journal of Network Security, vol.5, no.2, pp.145-153, Sept. 2007.

[33] J. Heidemann, F. Silva, C. Intanagonwiwat, R. Govindan, D. Estrin, and D. Ganesan,”Building efficient wireless sensor networks with low-level naming”, *In Proc. of the ACM Symposium on Operating Systems Principles,* Banff, Canada, (SOSP), Oct. 2001, pp. 146-159.

[34] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, and F. Silva,“Directed Diffusion for Wireless Sensor Networking,” ACM/IEEE Transactions on Networking (TON), vol.11 , no.1, pp. 2-16, Feb. 2003.

[35] C. K. Wai, "Advanced Approach for Directed Diffusion and Network Coding”, [Online], Available: http://personal.ie.cuhk.edu.hk/~kwwei/FYP/NC_DD.pdf

[36] B. M. Vetter, F. Wang, and S. F. Wu,“An experimental study of insider attacks on the OSPF routing protocols”, *In The 5th IEEE International Conference on Network Protocol*s, Atlanta, GA, (ICNP), Oct.1997, pp. 293-300 28-31

[37] E. Fasolo, M. Rossi, J. Widmer, and M. Zorzi,“In-network aggregation techniques for wireless sensor networks: a survey,” IEEE Transaction Wireless Communication., vol. 14, no. 2, pp. 70-87, Apr. 2007

[38] VR Kumar, J Thomas, A Abraham“Secure Directed Diffusion Routing Protocol for Sensor Networks using the LEAP Protocol", NATO Security through Science Series - D: Information and Communication Security, vol.6, pp. 183-203, 2006.

[39] R. D. Pietro, L. V. Mancini, Y. W. Law, S. Etalle, and P. J. M,“ Havinga, LKHW: A Directed Diffusion-Based Secure Multicast Scheme for Wireless Sensor Networks”, *32*nd *Int. Conf. on Parallel Processing Workshops,* Kaohsiung, Taiwan, (ICPP), IEEE Computer Society Press, Oct., 2003, pp. 397-406.

[40] P. Ning, K. Sun,“How to Misuse AODV: A Case Study of Insider Attacks against Mobile Ad-hoc Routing Protocols,” in Ad Hoc Networks, vol. 3, no. 6, pp. 795-819, Nov. 2005.

[41] VL Chee, WC Yau ," Security analysis of TORA routing protocol”, in Springer, vol. 4706, pp.975-986, August 2007.

[42] A. Kalambur,“Secure Routing in Wireless Sensor Networks: A study on Directed Diffusion”. Available: http:// ww.cs.sjsu.edu

[43] S. Moon, T. Cho,“Intrusion Detection Scheme against Sinkhole Attacks in Directed Diffusion Based Sensor Networks”, IJCSNS International Journal of Computer Science and Network Security, vol.9, no.7, pp. 118-122, Jul. 2009.

[44] J. Kim, P. Bentley, C. Wallenta, M. Ahmed, and S. Hailes, "Danger Is Ubiquitous: Detecting Malicious Activities in Sensor Networks Using the Dendritic Cell Algorithm," *Proc. of 5th International Conference on Artificial Immune Systems*, Oeiras, Portugal, (ICARIS), pp. 390-403, 2006.

[45] X. Wang, L. Yang, K. Chen, "SDD: Secure Directed Diffusion Protocol for Sensor Networks", Lecture Notes in Computer Science, vol. 3313, Jan. 2005, pp. 205-214.

[46] H. Yang, SH. Wong, S. Lu, L. Zhang, "Secure Diffusion for Wireless Sensor Networks", *3rd International Conf. on Broadband Communication, Networks, and Systems,* San Jose, CA, (BROADNETS) Oct. 2006.

[47] S. H. Chi and T. H. Cho, "Fuzzy Logic Anomaly Detection Scheme for Directed Diffusion Based Sensor Networks," Lecture Notes in Artificial Intelligence, vol.4223, Sep.2006, pp. 725-734.

[48] Y. Sun, Z. Han, K.R. Liu, "Defense of Trust Management Vulnerabilities in Distributed Networks", *IEEE Communications Magazine*, Feb. 2008, pp. 112-119.

[49] O. Younis and S. Fahmy, "Distributed Clustering in Ad hoc Sensor Networks: A Hybrid, Energy-Efficient Approach", *The 23rd Conf. of the IEEE Communications Society,* Hong Kong, (IEEE INFOCOM), Mar. 2004.

[50] W. B. Heinzelman. "Application-Specific Protocol Architectures for Wireless Networks", PhD dissertation, Massachusetts Institute of Technology, Jun. 2000.

[51] A. Cerpa and D. Estrin, "Ascent: Adaptive self-configuring sEnsor networks topologies. *In Proc. of the The 21st Annual Joint Conference of the IEEE Computer and Communications Societies,* New York, NY, USA, (IEEE INFOCOM), June 2002.

[52] L. Eschenauer and V. D. Gligor, "A key-management scheme for distributed sensor networks", *In Proc. of the 9th ACM Conference on Computer and Communications Security,* Washington, DC, USA, (CCS '02), Nov. 2002.

[53] A. Ferrante, R. Pompei, A. Stulova, , A.V. Taddeo, " A protocol for pervasive distributed computing reliability", *In the Proc. of the 4th IEEE International Conference on Wireless and Mobile Computing, Networking and Communication,* Avignon, France, (WiMob 2008) , Oct. 2008, pp. 574-579

[54] L.F. Perrone and S.C. Nelson, "A study of on-off attack models for wireless ad hoc networks," *1st IEEE International Workshop on Operator-Assisted (Wireless Mesh) Community Networks*, Berlin, Germany, (OpComm2006), Sep. 2006.

[55] M. G. Hinchey, R. Sterritt, and C. Rouff, “Swarms and Swarm Intelligence”, IEEE Computer Society, vol. 40, issue 4, April 2007, pp. 111-113.

[56] M.J. Warren, M. Dougall, K. Pascoe (2002), "Swarming attacks and agents”, [Online], Available: http://igneous.scis.ecu.edu.au/proceedings/2002/papers_full/26.pdf.

[57] P. Yi, Z. Dai, S. Zhang, Y. Zhong, "A New Routing Attack in Mobile Ad Hoc Networks”, *International Journal of Information Technology (IJIT),* vol. 11, no. 2, 2005, pp. 83-94.

[58] B. Awerbuch, R. Curtmola, D. Holmer, C. NitaRotau, and H. Rubens, (2004, March), “Mitigating Byzantine Attacks in Ad Hoc Wireless Networks” , Technical Report Version 1, Department of Computer Science, Johns Hopkins University, Baltimore, USA. [Online]. Available: http://www.citeseerx.ist.psu.edu

[59] A. Pathan, H. Lee, C. Hong, “Security in Wireless Sensor Networks: Issues and Challenges”, *In Proc. of 8th Advanced Communication Technology 2006,* Phoenix Park, Republic of Korea, (IEEE ICACT), vol. 2, no. 6, , 20-22 Feb. 2006 pp. 1048-1054.

[60] Y. W. Law, P. J.M. Havinga, “How to Secure a Wireless Sensor Network”, *In Proc. 2005 2nd International Conference on Intelligent Sensors, Sensor Networks and Information Processing Conference*, Melbourne, Australia (ISSNIP 2005), 5-8 Dec. 2005, pp. 89-95.

[61] T. Roosta, S. Shieh, and S. Sastry, ‘Taxonomy of security attacks in sensor networks,” *in 1st IEEE International Conf. on System Integration and Reliability Improvements*”, Hanoi, Vietnam, (SIRI'06), vol. 1, 2006, pp. 529-536.

[62] A. Perrig, J. Stankovic, D Wagner,” Security in wireless sensor networks”, *Communications of the ACM* (CACM), vol. 47, no. 6, June 2004, pp. 53-57.

[63] H. Chan, A. Perrig, “Security and Privacy in Sensor Networks”, *IEEE Computer*, vol.36, no. 10, 2003, pp. 103-105.

[64] A. Preeti, Y. Chaba, and Y. Singh, “Review of Detection and Prevention Policies for Distributed Denial of Service Attack in MANET”, *Proc. of 2nd National Conference on Challenges & Opportunities in Information Technology*, Mandi, Gobindgarh. (COIT-2008), March 29, 2008.

[65] S. Datema, “A Case Study of Wireless Sensor Network Attacks”, M.S. Thesis in Comp. Science, Fac. Elec. Eng., Math., and Comp. Science, Delft Univ. of Technology September 22th, 2005.

[66] E. Shi, A. Perrig, ‘Designing secure sensor networks”, *IEEE Wireless Communications*, vol.11, no. 6, December 2004, pp.38-43.

[67] V. Siris , F. Papagalou, “Application of anomaly detection algorithms for detecting SYN flooding attacks”, *Computer Communications*, vol.29 ,2006, pp.1433–1442.

[68] C. Intanagonwiwat, R. Govindan, and D. Estrin, ‘Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks,” *Proc. 4th ACM International Conference on Mobile Computing and Networking* (MOBICOM'98), Aug. 2000.

[69] F. J. Ros and P. M. Ruiz, “Implementing a new manet unicast routing protocol in ns-2,” Dept. of Information and Communications Engineering University of Murcia, Tech. Rep., 2004.

[70] K. Fall, K. Varadhan, “The ns Manual (formerly ns notes and documentation), the VINT project, July 2003.