

Swarm Flooding Attack against Directed Diffusion in Wireless Sensor Networks

Ibrahim S. I. Abuhaiba¹, Huda B. Hubboub

P. O. Box 108, Computer Engineering Department, Islamic University, Gaza, Palestine

¹isiabuhaiba@gmail.com

Abstract — The objective of this paper 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.

We present a new attack, Swarm Flooding Attack, against Directed Diffusion based WSNs, which targets the consumption of sensors computational resources, such as bandwidth, disk space, or processor time. Two variants of swarm attack have been introduced: 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. For each of the proposed attack models, we present analysis, simulation, and experimental measurements. We show that the system achieves maximal damage on system performance represented by many metrics.

Index Terms — wireless sensor network, denial of service attack, directed diffusion, swarming, flooding

I. INTRODUCTION

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 applications, 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.

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. As a sensory network protocol, Directed Diffusion is subject to many threats and risks. 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).

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 attack threats with no implementations of these attacks as it was the case in [3] and [4] where both papers investigate different misuse actions manipulated to attack AODV and TORA, respectively, to achieve certain attack objectives.

In [5], 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 [6], 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, Kalamhour [7] 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.

Reference [8] 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 [9], 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.

One category of attacks, flooding attacks, exploit the three-way handshake mechanism in TCP/IP protocol and are not applied to networks. Only one work introduced Ad Hoc Flooding Attack [10] briefly to attack a network running AODV protocol and did not explore any specifications of the attack. Also, only one work [11] theoretically mentions using the concept of swarming in attacking the web servers.

The main contribution of this work is the introduction of a new DoS attack, swarm flooding attack, framework against Directed Diffusion (DD) based WSN. This attack is 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. Our attack 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 participating in the attack. Ant attack itself has more than one version. All of the proposed distinct attacking techniques result in significant degradation in system performance.

The contributions of this research are highlighted hereunder:

- 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 new attack against Directed Diffusion based WSN, which can be applied to any other routing protocol.
- We investigate the impact of different forms of this attack which is implemented on NS-2 simulator. 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.

The paper is organized as follows. Our proposed attack is presented in section II. Experimental results are reported in section III. Finally, the paper is concluded in section IV.

II. PROPOSED DOS ATTACK AGAINST DD

A. 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 injecting 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. Our swarm flooding attack depends on traffic injection; two forms of this attack are discussed (Bee and Ant).

B. 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 nodes 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 queries (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.

C. Design Considerations

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 model produces maximal damage on the performance of the network.

D. Attack Goals

To successfully attack the network, our model has three goals: (1) compromise some of legitimate sensors and modify their 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. We explain these three goals as follows.

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.

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 control 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.

Hierarchical algorithms intensively present different formulas in order to divide the network into cells. In [13], 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 by the relation:

$$k = \left\lceil \frac{A}{\Pi \times r^2} \right\rceil \quad (1)$$

Another method to compute the optimal number of cells in a sensor network was presented in [14] where the optimal number of k cells is obtained using, n , the number of nodes, d , the distance to BS, s_{friss} and $s_{two-ray}$, the radio energy parameters. Then, attackers' number is given by:

$$k = \left\lceil \frac{\sqrt{n}}{\sqrt{2\Pi}} \sqrt{\frac{s_{friss} \text{ amp } M}{s_{two-ray} \text{ amp } d_{ioBS}^2}} \right\rceil \quad (2)$$

Although the aforementioned references can divide the network into 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 [15]. While the relation originating in [14] 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 contain such that the attacker in any cell could communicate with the maximum number of nodes within the same zone.

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 [16]. Assuming p is the probability that a link exists between two sensor nodes, n is the number of network nodes, d , being the expected degree of a node (i.e., the average number of edges connecting that node with its graph neighbors), equals to:

$$d = p(n-1) \quad (3)$$

We need to find out the value of d 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, P_c , for graph connectivity, the threshold function p is defined by the following formulas presented in [16]:

$$P_c = \lim_{n \rightarrow \infty} P_r [G(n, p) \text{ is connected}] = e^{-c} \quad (4)$$

$$p = \frac{\ln(n)}{n} + \frac{c}{n} \quad (5)$$

Therefore, given P_c , we can find c (real number), and with the knowledge of n , 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 (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:

$$k = \left\lceil \frac{n}{d} \right\rceil \quad (6)$$

Table I below contains the number of o attackers calculated from (1), (2), and (6), for different number of network size n , and network size is $100 \times 100 \text{ m}^2$.

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

Network size	Reference [13]	Reference [14]	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	$A = 100 \times 100 \text{ m}^2$ $r = 25 \text{ m}$	$S_{friss \text{ amp}} = 10 \text{ pJ}$ $S_{two-ray \text{ amp}} = 0.0013 \text{ pJ}$ $M = 100 \text{ m}$ $75 < d_{toBS} < 185$	$P_c = 0.99999$

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 would not 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 (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. Fig. 1 below demonstrates the division of the network into k attack zones where k equals the number of attackers calculated from (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.

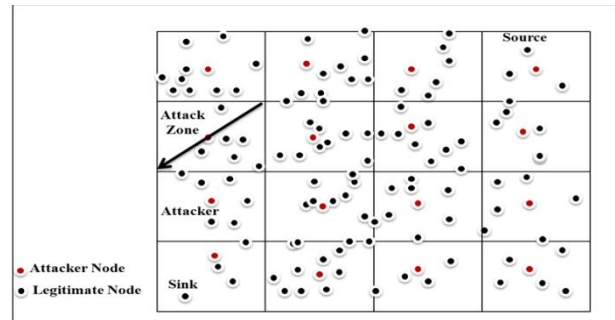


Figure 1: Attacker distribution into attack cells throughout the network

E. Attack Model

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 the consumption of computational resources, such as bandwidth, disk space, or processor time.

Swarming

The concept of swarming originated from nature. It is a general term that can be applied to any animal that swarms. The term applies particularly to insects; hive or nesting organizations such as ants or bees are the most familiar pattern for swarming [17]. 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 [11], 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 a related framework, there is flooding attack which involves sending large volume of traffic to a victim system, to congest the victim system's network bandwidth. The victim system slows down, crashes, or suffers from saturated network bandwidth, preventing access by legitimate users. 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 physical world is the difference between swarming strategies of Bee and Ant. So, we introduced two forms of flooding attack namely Bee Swarm Attack and Ant Swarm Attack. 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, which are inserted into the network until the system is saturated. Fig. 2 demonstrates the idea of flooding the network with fake interests and compares 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.

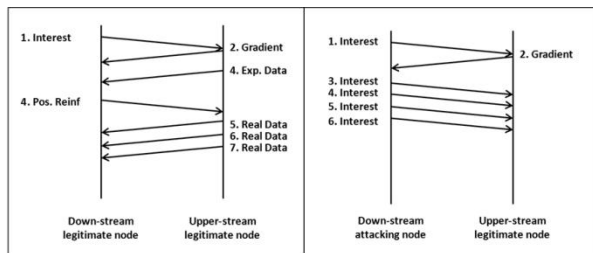


Figure 2: Representation of legal interest packet and the corresponding interactions in normal DD environment (left) compared to fake interest packet flooding in adversarial DD environment (right)

Bee Swarm Attack

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

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 three variants of ant attack:

- **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 are flooded into the network.
- **Forward hierarchical attack:** where the attackers enter the network in an increasing swarms or bursts. Each swarm consists of a different number of attackers and injects new interests' type into the network. The effect of the attack is expected to be earlier compared to the sequential one.
- **Reverse hierarchical attack:** where the shape of ant hierarchical order is reversed to allow larger bursts of attackers to be earlier in the network. The attack effect would be faster than both forward hierarchical and sequential attacks.

III. SIMULATION AND RESULTS

A. Simulation Setup and Implementation Details

We have used the Network Simulator (NS-2.32) [18, 19] to simulate a wireless sensor network running the Directed Diffusion routing protocol. 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. Simulation parameters were chosen in accordance with [2] and listed in Table II.

To verify our attack against Directed Diffusion, we implemented it in NS-2.32. 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. 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.

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

Parameter	Value
Simulation time	1300, 1500 second
Simulation area	800m × 800 m
Number of nodes	30
Transmission range	250 m
Link bandwidth	1.6 Mbps
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

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. 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 many metrics to create a simulation scenario that meets the researcher's target values for these metrics to a close approximation.

B. Performance Metrics

We choose the following metrics to measure the efficiency of our work:

- **Throughput:** It is the sum of received packets at sink, calculated at every time interval and divided by its length. 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.
- **Packet delivery ratio:** ratio of the packets delivered to the sink to those generated by the sources.
- **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.
- **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.

- **Routing overhead:** This measures the efficiency of the routing protocol. It is defined as the ratio between the total number of control packets transmitted to data packets. Control packets include route requests, replies and error messages.
- **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.
- **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 on the source node.

C. Simulation Results of Bee Swarm Flooding Attack

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. In Fig. 3, we observe that throughput 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 cannot get to the goal and those packets are discarded by nodes for network congestion. Interestingly, the network seems to have some recoverability.

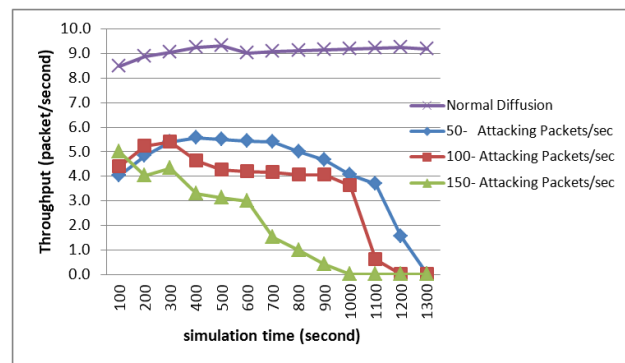


Figure 3: Effect of different attacking packet number on sink throughput over time

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 cannot bear the attack anymore and the performance goes down quickly.

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 III 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 Fig. 3, 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 III: System performance over different combinations of three factors, attacking packets = 250

Attackers	Interests	Interest rate	T_{Deny}	PDR	Throughput
5	1	50	1123	88.31%	3.86
1	5	50	1122	65.79%	3.84
8	30	1	923	51.23%	2.93

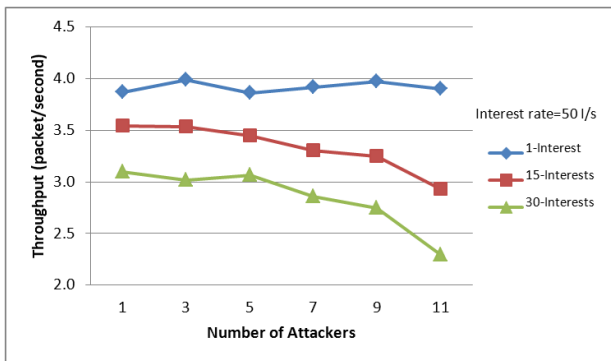


Figure 4: Throughput of different number of interest when changing number of attackers

Performance of Different Number of Interests when Changing Number of Attackers

Fig. 4 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). As interest entries in the cache do not 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 a 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} , Fig. 5. 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.

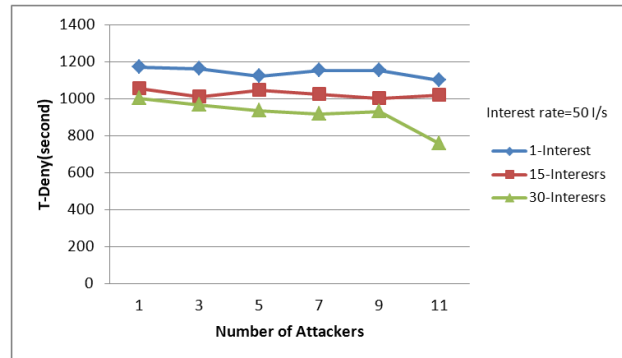


Figure 5: Deny time of different number of interest when changing number of attackers

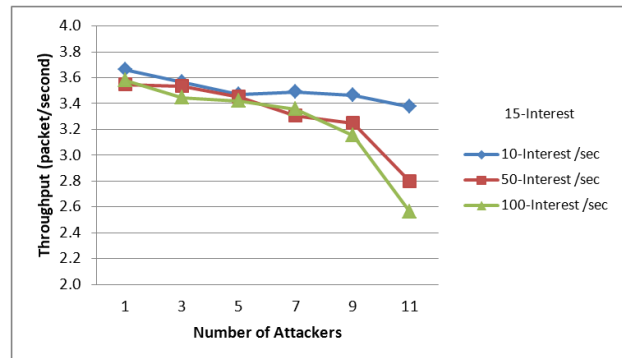


Figure 6: Throughput of different interest rate when changing number of attackers

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. Fig. 6 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 a 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 $11 \times 100 \times 15$ 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.

Fig. 7 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.

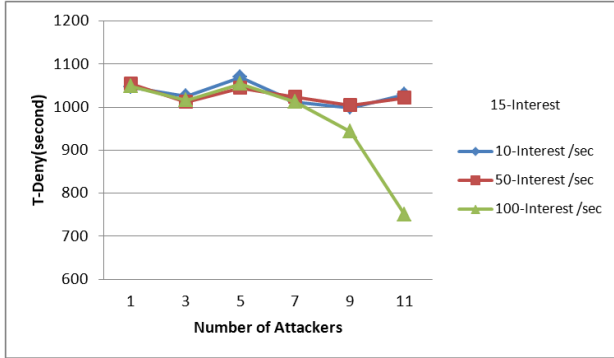


Figure 7: Deny time of different interest rate when changing number of attackers

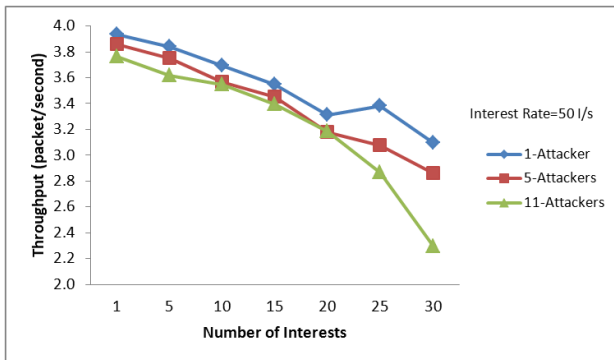


Figure 8: Throughput of different attackers' number when changing number of interests

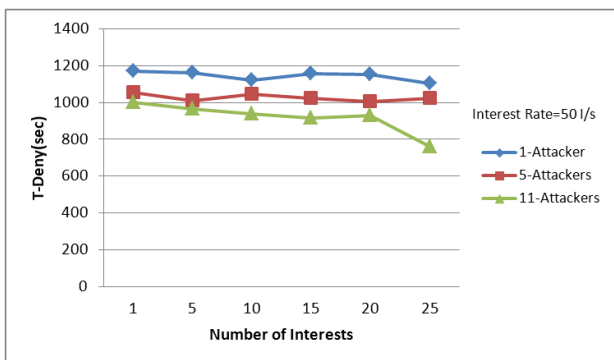


Figure 9: Deny time of different attackers' number when changing number of interests

Performance of Different Attackers' Number when Changing Number of Interests

Figs. 8 and 9 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. 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.

Performance of Different Interest Rate when Changing Number of Interests

Fig. 10 proves that the number of interests is the main factor that influences the throughput regardless the value of the data rate.

In addition, as Fig. 11 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 time in processing incoming packets.

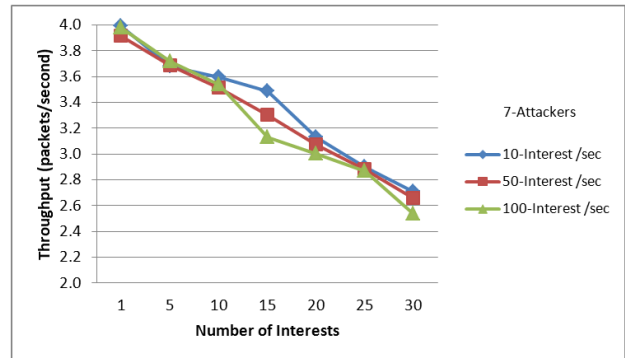


Figure 10: Throughput of different interest rate when changing number of interests

Performance of Different Attackers' Number when Changing Interest Rate

Although we have shown that interest rate does not affect the network behavior, more investigation would show some effect. These effects are not visible in previous figures. It is easily observed from Fig. 12 that the behavior of the three curves is close.

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 Fig. 13.

Performance of Different Interests' 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 Fig. 14 and Fig. 15. 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.

D. 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.

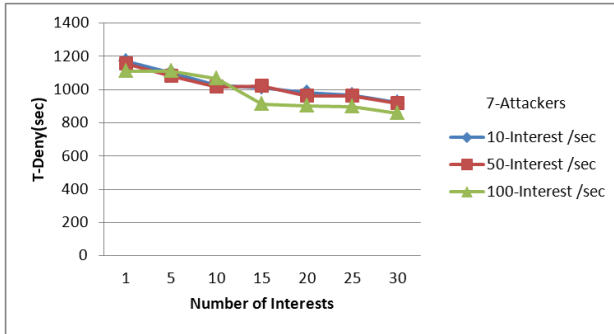


Figure 11: Deny time of different interest rate when changing number of interests

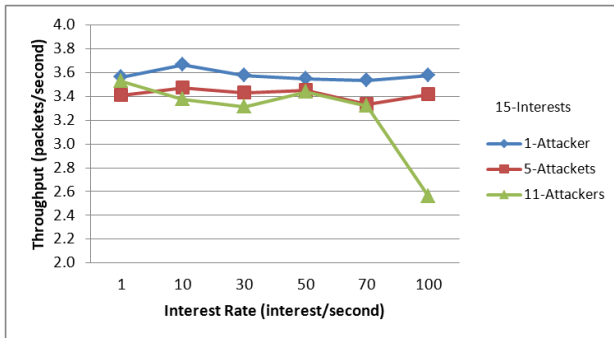


Figure 12: Throughput of different attackers' number when changing interest rate

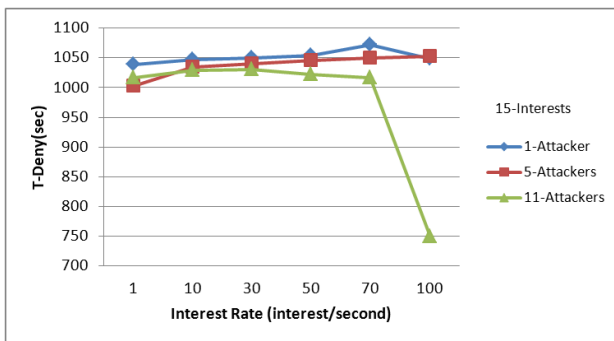


Figure 13: Deny time of different attackers' number when changing interest rate

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.

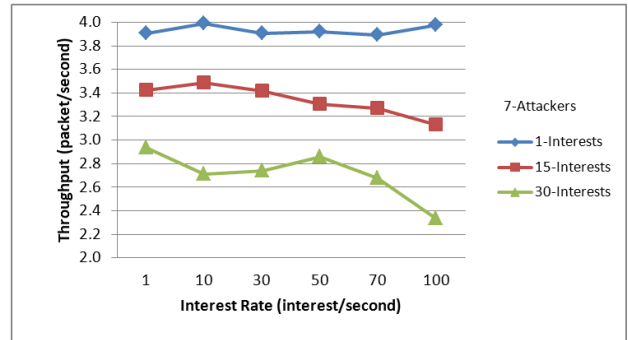


Figure 14: Throughput of different interests' number when changing interest rate

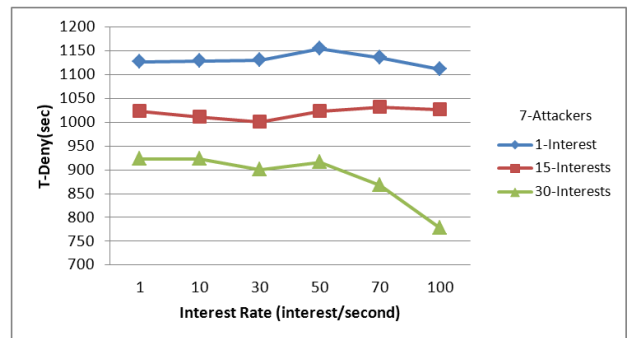


Figure 15: Deny time of different interests' number when changing interest rate

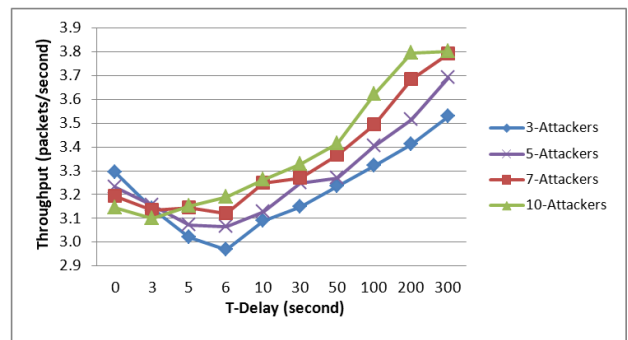


Figure 16: Performance of ant sequential attack in term of sink throughput

Fig. 16 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 seconds all the 10 attackers were available in the network. However, the 3 attackers would be completely effective at 300 seconds. Notice 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.

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 IV demonstrates the bursts of both types of attack.

Table IV: 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 Fig. 17 and Fig. 18. At first glance, one may think that the 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 Fig. 18 compared to Fig. 17 and earlier in Fig. 17 compared to Fig. 16. 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.

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, notice that the amplitude of the throughput decreases in reverse hierarchical compared to the hierarchical.

Comparison of Different Swarm Attacks

We conducted another simulation to compare among these schemes. Fig. 19 and Fig. 20 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.

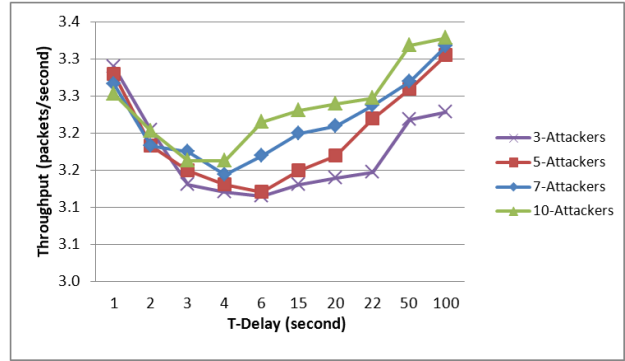


Figure 17: Performance of top-to-base hierarchical ant attack in term of sink throughput

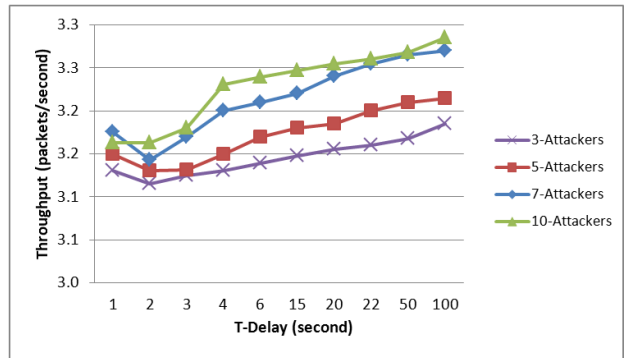


Figure 18: Performance of base-to-top hierarchical ant attack in term of sink throughput

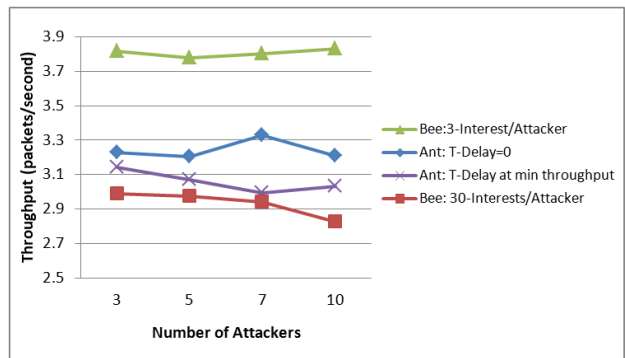


Figure 19: Comparison of different swarm attacks in term of sink throughput

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 Fig. 19, 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 (Fig. 21 and Fig. 22) show that bee outperforms ant in increasing the packet delivery delay noting the difference in interest number disseminated by each attacker in both cases.

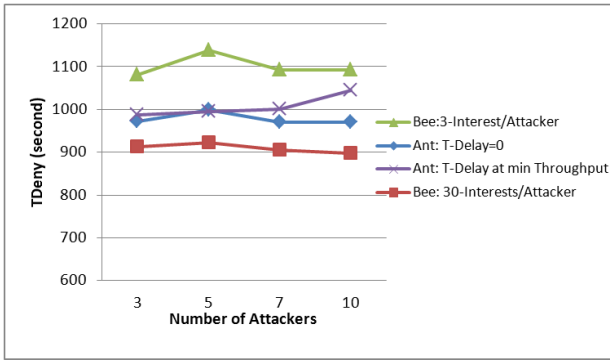


Figure 20: Comparison of different swarm attacks in term of sink deny time

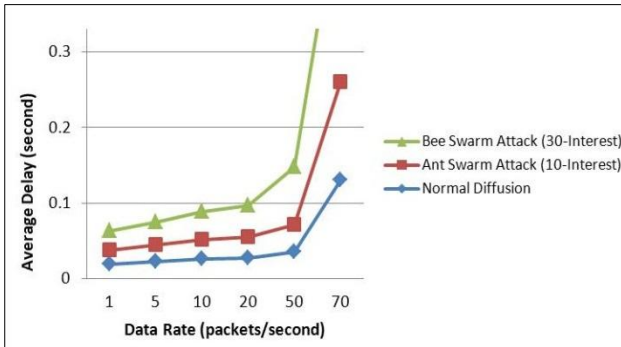


Figure 21: Comparison of different swarm attacks in term of average delay with small data rate

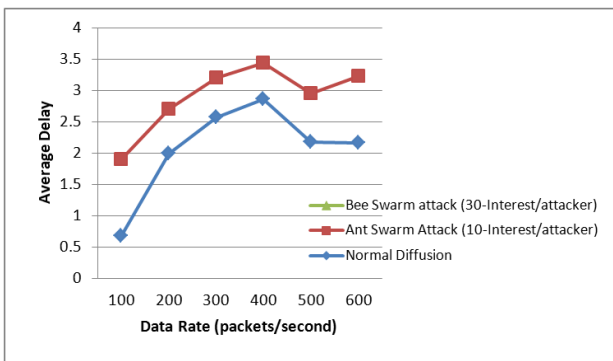


Figure 22: Comparison of different swarm attacks in term of average delay with high data rate

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 Fig. 23 as the victim network is similar to the normal diffusion but with less throughput values. Fig. 24 also indicates that our attack decreases the number of data sent by source.

We plot T_{deny} as a function of number of sinks in Fig. 25. 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.

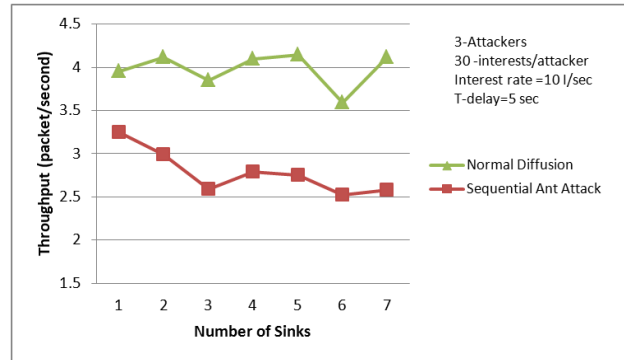


Figure 23: Performance of sequential ant attack over multiple sinks network

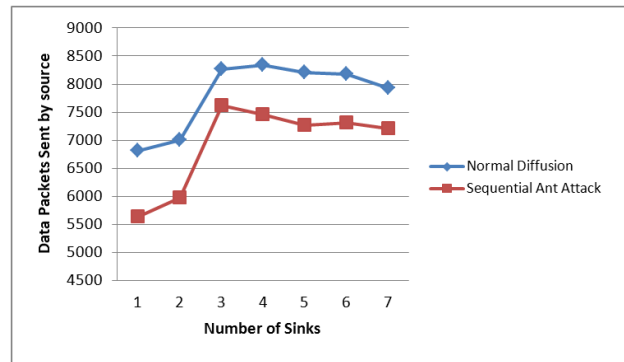


Figure 24: Number of data packets sent by source in multiple sinks network

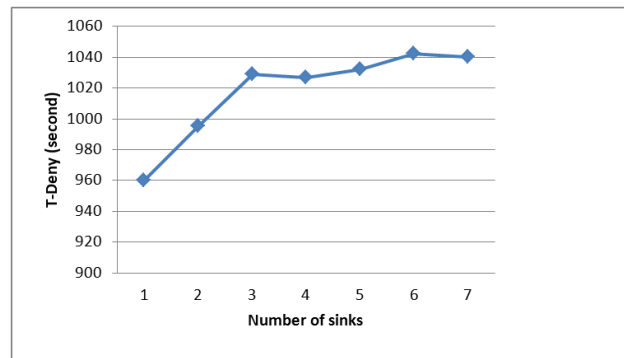


Figure 25: Deny time under sequential ant attack in multiple sinks network

E. Discussion

In this section, we presented 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 [20], the authors defined relative strength of a particular attack configuration Σ , which represents the amount of damage an attack can cause per adversary, as:

$$\Sigma = \frac{DR_{norm} - DR_{adv}}{DR_{norm} \cdot Num_{adv}}, \quad (7)$$

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 V.

Table V: Attack strength for the simulated attacks

Attack type	Attack strength
Bee swarm attack	13.98
Sequential ant swarm attack	12.38
Hierarchical ant swarm attack	12.53
Reverse hierarchical ant swarm attack	12.59

The results indicate relatively high attack strength compared to values obtained in [20]; 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.

IV. CONCLUSION

This paper 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.

We have introduced a new attack against DD based WSN, namely, Swarm Flooding Attack. This attack allows an attacker to mount a DoS attack against most of currently proposed on-demand routing protocols. This attack integrates the concepts of swarming and flooding. It 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 proved that attacking a target from many locations could be done in different ways. Bee Swarm Attack is the first model which we validated 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 the 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 Swarm, 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.

ACKNOWLEDGMENT

We thank anonymous referees for their constructive comments.

REFERENCES

- [1] A. D. Wood, J. A. Stankovic, "Denial of service in sensor networks," *IEEE Computer*, Vol. 35, pp. 54-62, 2002.
- [2] C. Intanagonwiwat, R. Govindan, D. Estrin, "Directed diffusion: a scalable and robust communication paradigm for sensor networks", *MobiCom 2000*, pp. 56-67, 2000.
- [3] P. Ning, K. Sun, "How to misuse AODV: A case study of insider attacks against mobile ad-hoc routing protocols," *Ad Hoc Networks*, Vol. 3, pp. 795-819, 2005.
- [4] V. L. Chee, W. C. Yau, "Security analysis of TORA routing protocol," *Lecture Notes in Computer Science*, Vol. 4705, pp. 975-986, 2007.
- [5] C. Karlof, D. Wagner, "Secure routing in wireless sensor networks: Attacks and countermeasures," *Ad Hoc Networks*, Vol. 1, pp. 293-315, 2003.
- [6] V. R. 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.

- [7] A. Kalambur, "Secure routing in wireless sensor networks: A study on directed diffusion," available at <http://www.cs.sjsu.edu/~stamp/CS265/projects/Spr04/section1/papers/Kalambur.doc>, 2004.
- [8] S. Moon, T. Cho, "Intrusion detection scheme against sinkhole attacks in directed diffusion based sensor networks," *International Journal of Computer Science and Network Security*, Vol. 9, pp. 118-122, 2009.
- [9] J. Kim, P. Bentley, C. Wallenta, M. Ahmed, S. Hailes, "Danger is ubiquitous: Detecting malicious activities in sensor networks using the dendritic cell algorithm," *ICARIS 2006*, pp. 390-403, 2006.
- [10] P. Yi, Z. Dai, S. Zhang, Y. Zhong, "A new routing attack in mobile ad hoc networks," *International Journal of Information Technology*, Vol. 11, pp. 83-94, 2005.
- [11] M. J. Warren, M. Dougall, K. Pascoe, "Swarming attacks and agents," available at http://igneous.scis.ecu.edu.au/proceedings/2002/papers_full/26.pdf, 2002.
- [12] C. Hartung, J. Balasalle, R. Han, "Node compromise in sensor networks: The need for secure systems," *University of Colorado Technical Report CU-CS-990-05*, 2005.
- [13] O. Younis, S. Fahmy, "Distributed clustering in ad hoc sensor networks: A hybrid, energy-efficient approach," *INFOCOM*, pp. 629-640, 2004.
- [14] W. B. Heinzelman, "Application-specific protocol architectures for wireless networks," PhD thesis, Massachusetts Institute of Technology, USA, 2000.
- [15] A. Cerpa, D. Estrin, "Ascent: Adaptive self-configuring sensor networks topologies," *IEEE Transactions on Mobile Computing*, Vol. 3, pp. 272-285, 2004.
- [16] L. Eschenauer, V. D. Gligor, "A key-management scheme for distributed sensor networks," *CCS 2002*, pp. 41-47, 2002.
- [17] M. G. Hinchey, R. Sterritt, C. Rouff, "Swarms and Swarm Intelligence," *IEEE Computer Society*, Vol. 40, pp. 111-113, 2007.
- [18] K. Fall, K. Varadhan, "NS notes and documentation, the VINT project," available at http://www.isi.edu/nsnam/ns/doc/ns_doc.pdf, 2011.
- [19] P. Pancardo, J. C. Dueñas, "A proposal for system architecture to integrate scarce-resources wireless sensor networks into ubiquitous environments," available at <http://ftp.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-208/paper23.pdf>, 2006.
- [20] A. Pathan, H. Lee, C. Hong, "Security in wireless sensor networks: Issues and challenges", *ICACT 2006*, Vol. 2, pp. 1048-1054, 2006.

document analysis and understanding, pattern recognition, artificial intelligence, information security, and computer networks. Prof. Abuhaiba published tens of original contributions in these fields in well-reputed international journals and conferences.

Huda B. Hubboub received her B.Sc. degree in electrical engineering, Islamic University of Gaza, in 2002, and master degree in computer engineering, Islamic University of Gaza, in 2010. Her research interests include information security, computer networks, and digital image processing.

Ibrahim S. I. Abuhaiba is a professor at the Islamic University of Gaza, Computer Engineering Department. He obtained his Master of Philosophy and Doctorate of Philosophy from Britain in the field of document understanding and pattern recognition. His research interests include computer vision, image processing,