I. J. Computer Network and Information Security, **2013, 3, 13-24**
Published Online March 2013 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijcnis.2013.03.02

Reinforcement Swap Attack against Directed Diffusion in Wireless Sensor Networks

Ibrahim S. I. Abuhaiba[1], Huda B. Hubboub
P. O. Box 108, Computer Engineering Department, Islamic University, Gaza, Palestine
[1]isiabuhaiba@gmail.com

***Abstract* —** In this paper, we introduce a new attack, Reinforcement Swap Attack, against Directed Diffusion based WSNs, 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. For the proposed attack, 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, on-off attack

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 the 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 [7], Kalambour 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 have 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.

The main contribution of our work is the introduction of a new DoS 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 new attack is called 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.

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 investigate the impact of different forms of this attack which are implemented on NS-2 simulator. We demonstrate through simulation the effects of the presented attack on the Directed Diffusion routing protocol. Our results quantify the damage caused by the attack and provide insights into identifying those which result in the greatest network disruption while requiring the least number of adversarial participants.

The paper is organized as follows. Our proposed DoS 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. 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.

In our work, we tackle the first category to form DoS attacks. We target 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.

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 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.

C. *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.

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) attackers should be well distributed and organized in the network grid to achieve maximal damage. Details on how to reach these goals for swarm flooding attack against DD in wireless sensor networks was introduced in [10]. For the tack of this paper, these goals can attained in similar way.

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 on-off reinforcement swap attack which disrupts configuration information, such as routing information.

Reinforcement Swap Attack

Our attack is based on swapping the rule of routing signaling of Directed Diffusion. 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 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.

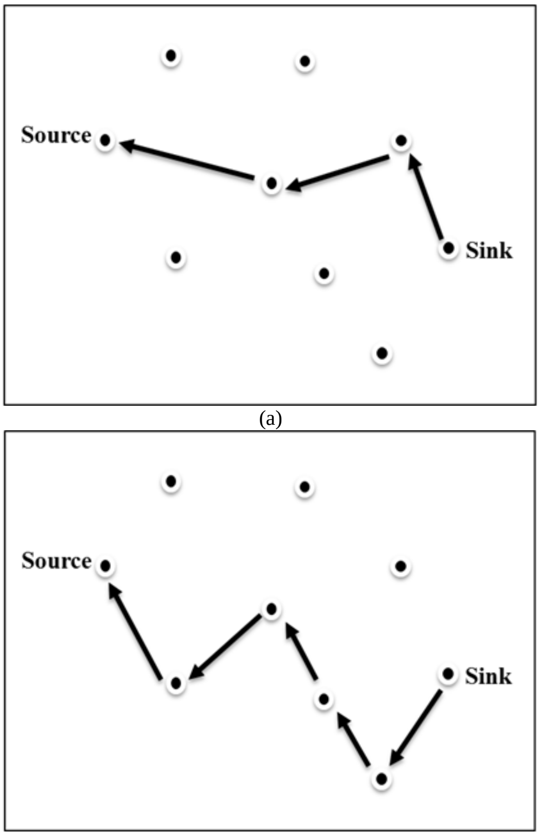
Sending Fake Negative

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 spreads 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 network allow the system to recover from the attack and search for alternative paths to deliver the required data. This feedback increases the probability of deleting better path along the network. We demonstrate our attack by comparing it with the original DD in two cases. Fig. 1 shows the first case where only one path is reinforced from sink to source. Fig. 1 (b) depicts that the swap attack, unlike original DD in (a), selects the longest path along the network. Also, in Fig. 2 where it represents the second case in which more than one path is reinforced, our attack reinforces the highest two delay paths.



(a)

(b)

Figure 1: The reinforced path in a single path establishment type DD based WSN: (a) normal operation of DD, (b) swap operation of DD

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:

Norm Mode: 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.

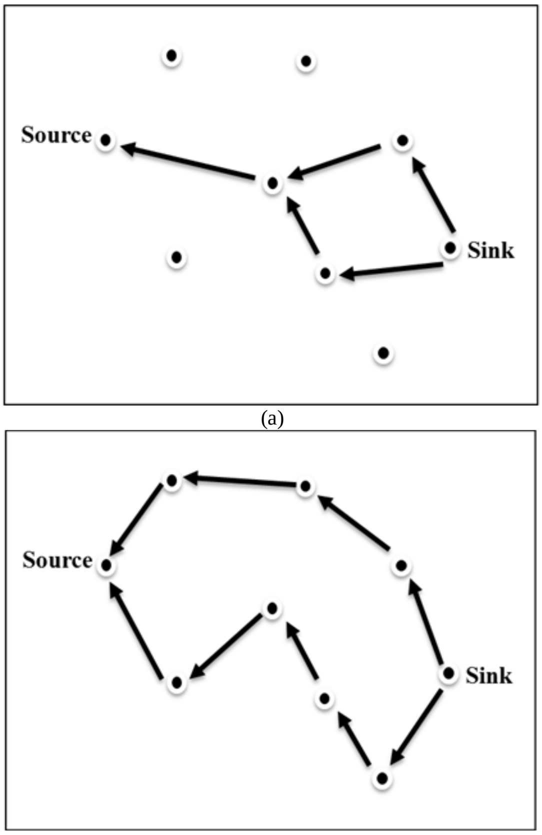
Halt Mode: where the attacker fluctuates between behaving bad and do nothing in the on/off cycles of the attack. Here, the effect of the attack is strong and total where the service is denied fast hence the system collapses earlier than the norm mode.

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.

On-Off Timer: 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 Fig. 3.

On-Off Counter: 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, Fig. 4.



(a)

(b)

Figure 2: The reinforced paths in multi-paths establishment type DD based WSN: (a) normal operation of DD, (b) swap operation of DD

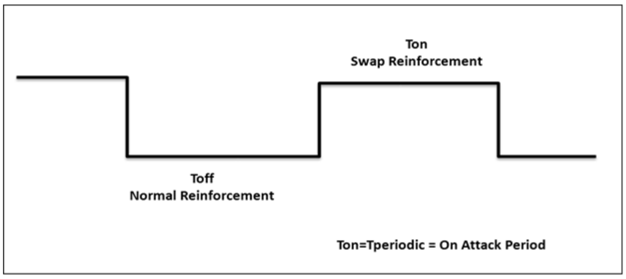


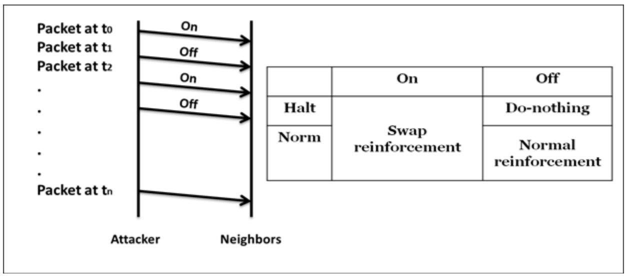
Figure 3: On/Off Attack periods of timer swap attack

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 [11].
- *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.



	On	Off
Halt	Swap reinforcement	Do-nothing
Norm		Normal reinforcement

Figure 4: On/Off Attack activations of counter swap attack

III. EXPERIMENTAL RESULTS

A. *Simulation Setup and Implementation Details*

For our experiments, we have used the Network Simulator (NS-2.32) [12] to simulate a wireless sensor network running the Directed Diffusion routing protocol. 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.

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 I.

Table I: 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 verify our attack in 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. 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.

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**.**

B. *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.

C. *Performance Metrics*

We used 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 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.
- *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 it has an impact on the source node.

D. *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.

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

In the design of our 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 Fig. 5 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 Fig. 5, 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.

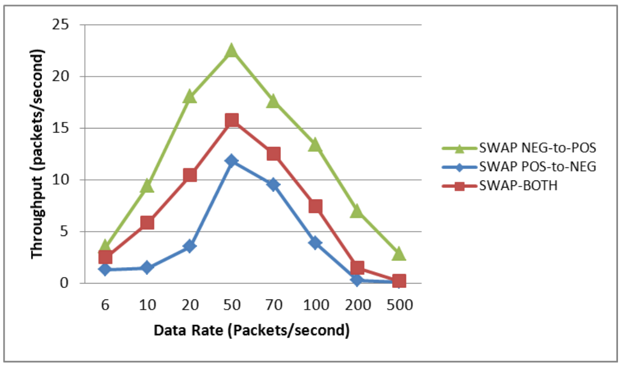


Figure 5: Effect of reinforcement swap anatomy on network throughput

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 Fig. 6 below.

The content of Fig. 6 is the same as Fig. 5 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 Fig. 5. 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 does not 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.

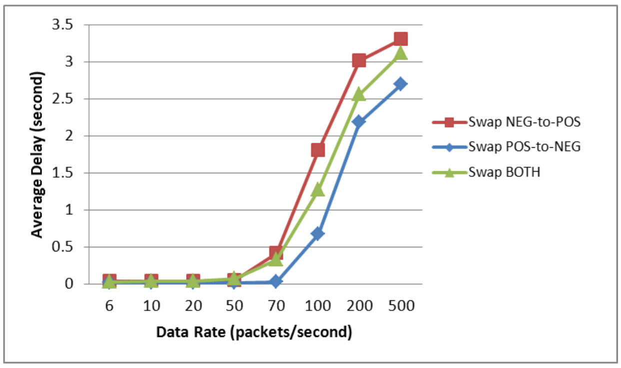


Figure 6: Effect of reinforcement swap anatomy on network average delay

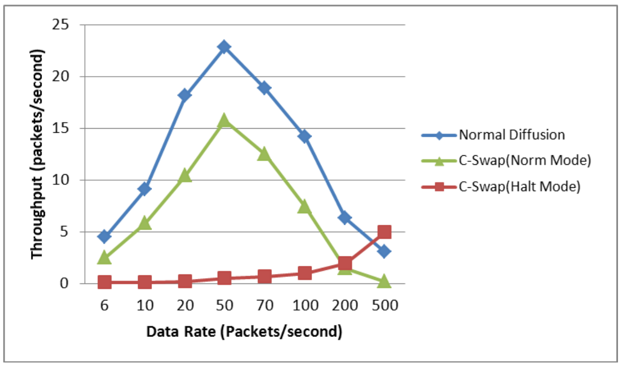


Figure 7: Effect of different swap attack modes on throughput while changing data rate

Performances of Different Swap Attack Modes while Changing Data Rate

Fig. 7 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.

We next performed another analysis to measure another essential metric which is the average delay. Fig. 8 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.

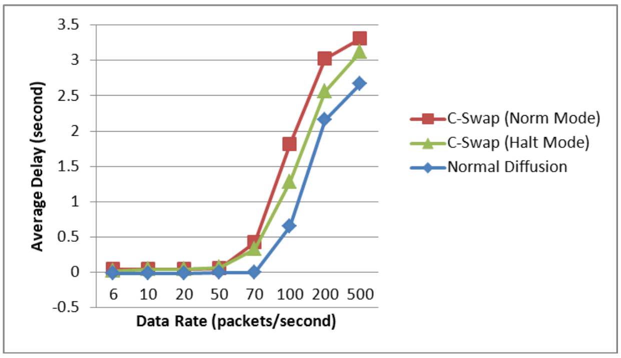


Figure 8 Effect of different swap attack modes on average delay while changing data rate

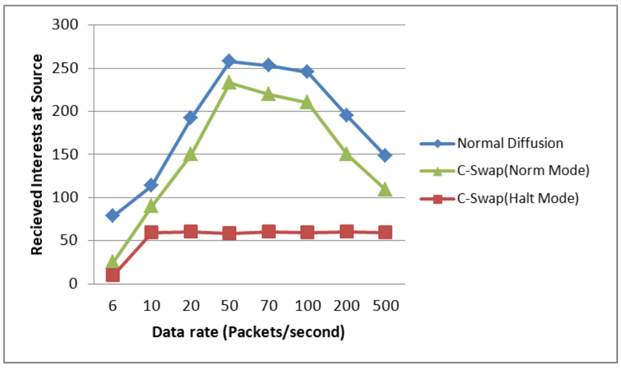


Figure 9: Effect of different swap attack modes on received interests at source while changing data rate

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 Fig. 9. As the figure indicates, there has been a decline in the received interests at the source.

Performance of Different Swap Attack Modes over Time

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. Fig. 10 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 cannot reach their target and those packets are discarded by nodes for network congestion, and the network cannot 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, Fig. 11. It is found in our study, as the graph can tell, that our attacker can produce no pronounced effects on this metric.

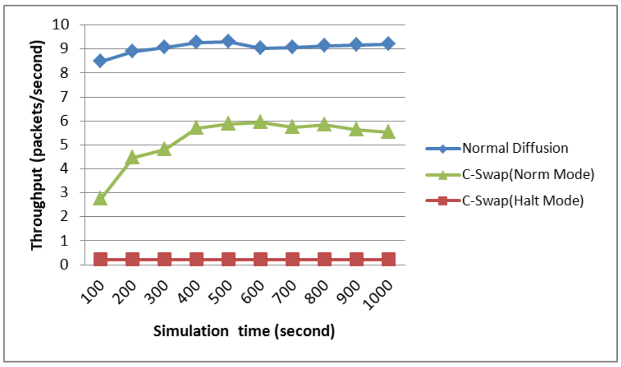


Figure 10: Throughput of different swap attack modes over time

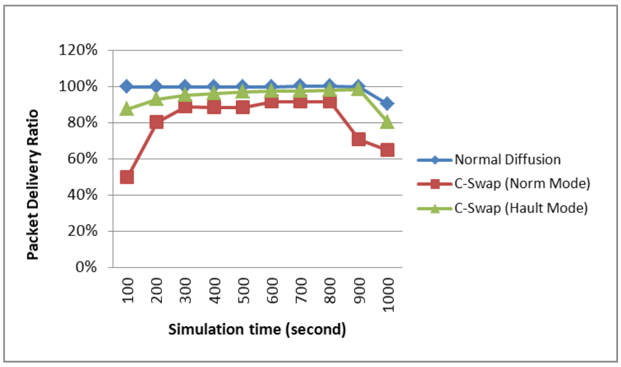


Figure 11: Packet delivery ratio of different swap attack modes over time

Packet delivery ratio is almost constant throughout time hence we cannot 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, Fig. 12, justifies this drop in the sent data at source node. We found that, during our attack, the number of received interests at source declines as compared to the case when original DD strategy is used for routing.

Fig. 12 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.

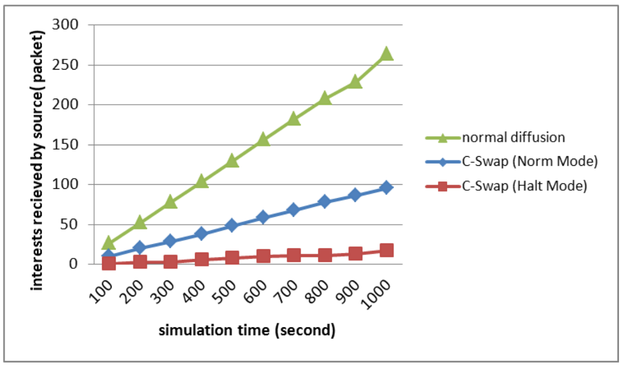


Figure 12: Received interests by source for different swap attack modes over time

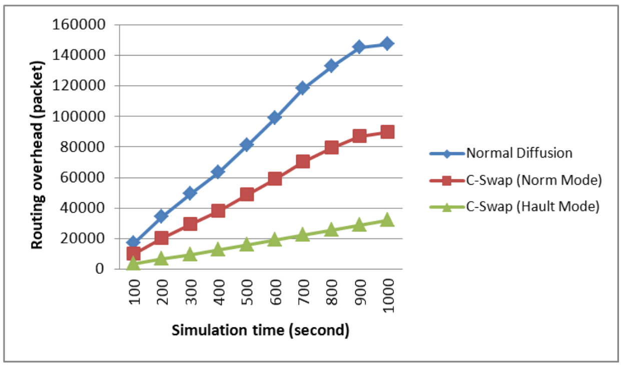


Figure 13: Routing overhead of different swap attack modes over time

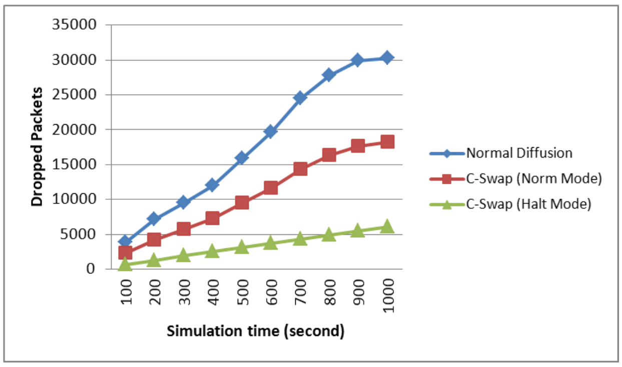


Figure 14: Dropped packets of different swap attack modes over time

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 Fig. 13 and Fig. 14. 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 does not try to retransmit broadcast packet in case there is a collision and the packet is simply dropped. Coupled with this fact, MAC-802.11 does not perform random selection of slots in the contention window before it transmits a packet [13].

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. Fig. 15 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.

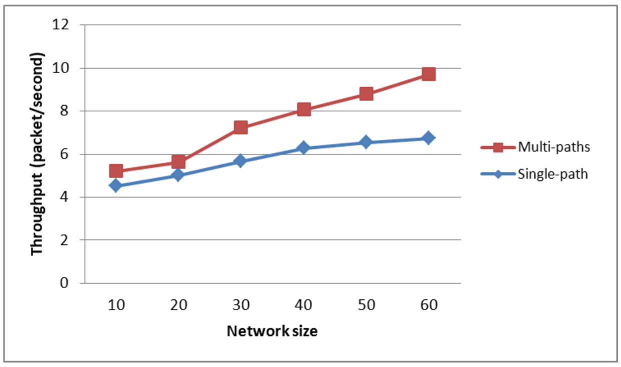


Figure 15: Performance of single/multi path(s) norm mode attack while changing network size

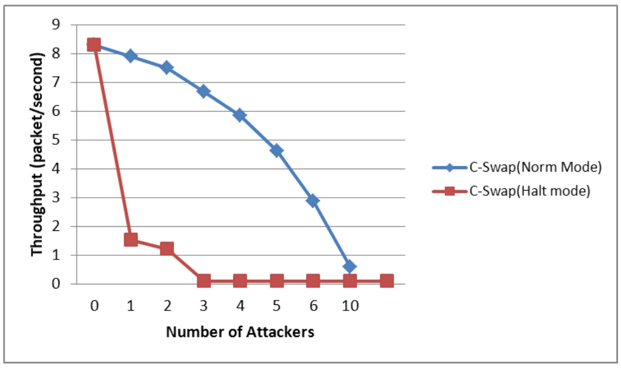


Figure 16: Performance of different attack modes when changing the number of attackers

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. Fig. 16 illustrates the simulation results with the configurations of up to 10 malicious nodes injected in regular distribution through the network.

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.

E. Simulation Results of Timer Reinforcement Swap Attack

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 Fig. 17, 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.

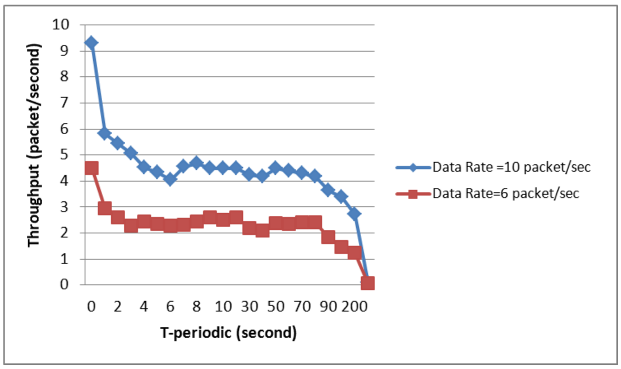


Figure 17: Performance of timer swap attack with changing attack period

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 Fig. 18, 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.

For the same parameters of Fig. 18, Fig. 19 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.

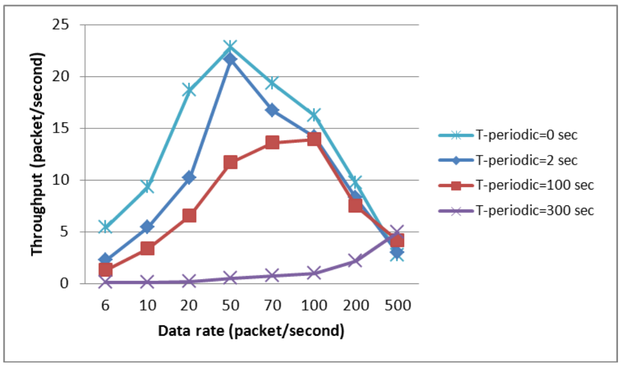


Figure 18: Throughput of timer swap attack with changing data rate

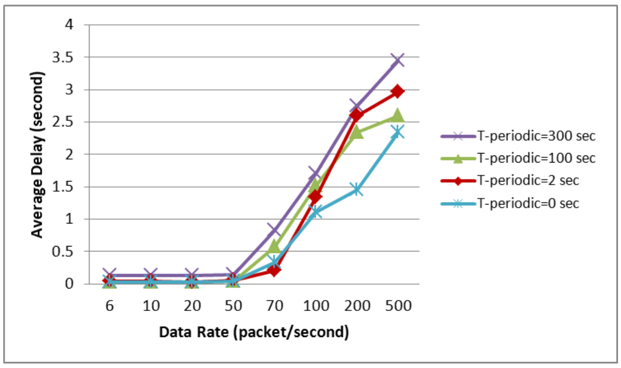


Figure 19: Average delay of timer swap attack with changing data rate

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 Fig. 20, 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.

The previous discussion is also convincing for Fig. 21 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.

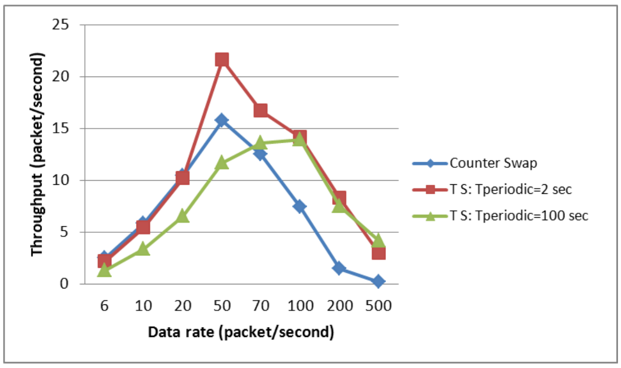


Figure 20: Comparison between counter and timer swap attack in term of sink throughput

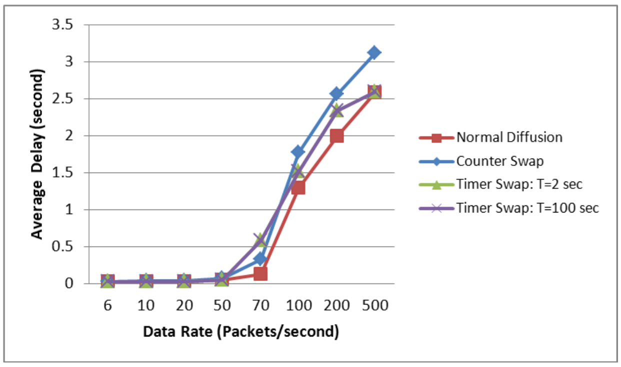


Figure 21: Comparison between counter and timer swap attack in term of average delay

F. 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 [14], the authors defined relative strength of a particular attack configuration Σ, which represents the amount of damage an attack can cause per adversary, as:

$$\Sigma = \frac{DR_{norm} - DR_{adv}}{DR_{norm} \cdot Num_{adv}}, \qquad (1)$$

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 II.

Table II: Strength of 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

The results indicate relatively high attack strength compared to values obtained in [14]; 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.

Throughout this paper, we have introduced a new attack against Directed Diffusion based WSN, namely Reinforcement Swap Attack. Swap Attack 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 presented 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 applications (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.

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.

REFERENCES

[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, 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] 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, no. 6, pp. 795-819, Nov. 2005.

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

[5] C. Karlof, D. Wagner, "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.

[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: http:// ww.cs.sjsu.edu

[8] 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.

[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," *Proc. of 5th International Conference on Artificial Immune Systems*, Oeiras, Portugal, (ICARIS), pp. 390–403, 2006.

[10] Ibrahim S. I. Abuhaiba, Huda B. Hubboub, "Swarm Flooding Attack against Directed Diffusion in Wireless Sensor Networks," International Journal of Computer Network and Information Security (IJCNIS), Vol. 4, No. 12, pp. 18-30, 2012.

[11] 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

[12] K. Fall, 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

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

[14] 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, Feb. 2006, pp. 1048-1054.

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, 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.