Research online Publishing www.ronpub.com

Open Access

Open Journal of Internet of Things (OJIOT)
Volume 5, Issue 1, 2019
http://www.ronpub.com/ojiot
ISSN 2364-7108

Understanding the Performance of Software Defined Wireless Sensor Networks under Denial of Service Attack

Gustavo A. Nunez Segura[A], Cintia Borges Margi[A], Arsenia Chorti[B]

[A] Universidade de São Paulo, Av. Prof. Luciano Gualberto, travessa 3, n.158, São Paulo, Brazil
{gustavoalonso.nunez, cintia}@usp.br
[B] ETIS, University Paris Seine, University Cergy-Pontoise, ENSEA, CNRS, 95000 Cergy-Pontoise, France,
arsenia.chorti@ensea.fr

Abstract

Wireless sensor networks (WSN) are formed from restricted devices and are known to be vulnerable to denial of service (DoS) security attacks. In parallel, software-defined networking has been identified as a solution for many WSN challenges with respect to flexibility and reuse. Conversely, the SDN control plane centralization may bring about new security threats and vulnerabilities. In this work, we perform a traffic analysis of software-defined WSN (SDWSN) in order to gain understanding of the network's performance when it is under certain types of DoS attacks. In particular, we consider three different DoS scenarios of increasing aggressiveness: (i) false flow requests DoS, (ii) false data flow forwarding DoS, and, (iii) false neighbor information passing DoS. Our simulation results for the latter two types of attack showed significant changes both in the average value and the variance of the delivery rate and the overall overhead. These results demonstrate that it is possible to identify when a SDWSN is under a particular type of DoS, by monitoring the respective quantities.

Type of Paper and Keywords

Regular research paper: *software-defined networking, traffic analysis, denial of service security attacks*

1 Introduction

Wireless sensor networks (WSN) are formed from restricted devices and their main purpose is to collect and process sensed data. WSN are widely used in environmental, industrial, and health monitoring applications, which turn them a key technology for Internet of Things (IoT). One of the main challenges faced by wireless sensor networks (WSN) and the Internet of Things (IoT) is security. WSN have certain resources and deployment characteristics that differentiate them from wired and non-resource-constrained networks, increasing their vulnerability to security attacks [21]. Also, WSN are commonly deployed in hostile environments, which increase the risk of physical attacks.

Software-defined wireless sensor networks (SDWSN) offer on the other hand solutions to many WSN challenges, in particular concerning flexibility and resource reuse [11]. The SDWSN approach decouples the control plane from the data plane and its main characteristic is the ability to program the network

This paper is accepted at the *International Workshop on Very Large Internet of Things (VLIoT 2019)* in conjunction with the VLDB 2019 conference in Los Angeles, USA. The proceedings of VLIoT@VLDB 2019 are published in the Open Journal of Internet of Things (OJIOT) as special issue.

operation dynamically [15][8]. The SDN controller's global view of the network could be instrumental in detecting the presence of malicious nodes or abnormal/suspicious behavior, based on the monitoring of the network traffic and/or other performance metrics [1]. Additionally, the SDN controller could take suitable countermeasures against a particular type of attack, e.g., by isolating malicious nodes, by using conditional rules to change how a device responds under certain conditions or to certain types of packets. On the other hand, the SDN's centralized architecture creates vulnerabilities to new security threats.

The planes' separation and the control decisions centralization in SDN renders the network prone to attacks that may not affect traditional networks. For example, SDN was originally designed as a single controller architecture, which in terms of security mounts to a single point of failure and thus increases vulnerability. In these conditions, an intruder may flood the network with control packets to exhaust the controller's resources, harming the network operation. A malicious node could also attack the network's controller by exploiting communication with legitimate network devices. As an example, an attacker could send messages with falsified information to its neighbors to force them to request new rules to the controller. This kind of attacks could harm both the controller's as well as the network devices' operation and performance.

So far, these new security issues, specific to SDWSNs, remain a largely uncharted area of research. Existing literature includes extensive works on intrusion detection in WSNs, but these analyzes and related proposed solutions are not suited for SDWSNs, as they do not consider the planes' decoupling and the SDNs' vulnerabilities [23][19]. On the other hand, intrusion detection proposals for SDN in wired networks [1][7] do not (down) scale to SDWSNs constrained networks, since the proposed approaches use more energy and bandwidth resources than a WSN could provide. As an example, Wang *et al.* [23] focus on routing in SDWSN, comparing their proposal to SDN-WISE when both networks are under attack. Authors focus on the selective forwarding attack and new flow requests. The first attack applies to any type of WSNs, while the second is specific for SDN.

In this work, we will attempt an analysis of the impact of three different DoS attacks on SDWSNs, aiming at shedding light on their impact on important network metrics; our ultimate goal is to exploit the findings of this study in future work on the early detection of such attacks in SDWSNs. In this framework and considering the SDWSN vulnerabilities described before, we implement three different DoS attacks: (i) a false flow request (FFR) attack, (ii) a false data flow forwarding (FDFF) attack, and, (iii) a false neighbor information (FNI) attack. The first one is similar to the new flow attack implemented by Wang *et al.* [23]. To improve the SDWSNs' resilience to DoS attacks one needs to understand the vulnerabilities that would allow malicious nodes to harm the network operation or performance. Therefore, our objective in this contribution is to provide the tools to understand the performance of a SDWSN when it is under these three attacks.

To achieve our goal, we executed simulations running IT-SDN [14] for six different square grid network sizes (36, 49, 64, 81, 100 and 121 nodes). We also considered two different sizes for the set of malicious nodes executing the attacks: a single attacker and approximately 10% of the network size. We analyze the following performance metrics: delivery rate, delay, control overhead and energy consumption. Results for the FDFF and FNI DoS attacks showed significant changes in the delivery rate, and in the overhead, both in terms of the average as well as the variance of these metrics.

Interestingly, in the FDFF attack, the impact on the average value is more accentuated than the impact on the variance of these metrics, while the inverse has been observed for the FNI attack. In future work, we will investigate the possible employment of lightweight change point anomaly detection methods [20] on the SDN controller, in order to identify changes in the average and/or the variance of these metrics. Ultimately, we envision to be able to flag not only the occurrence of an "abnormal" event, but, further provide an initial guideline regarding the nature of the underlying type of DoS attack.

The remaining of this paper is organized as follows. Section 2 provides a detailed description of the three attacks, while Section 3 explains set-up of our simulation platform for their implementation. The methods and experiments conducted are explained in Section 4. Section 5 presents the results and related discussion and Section 6 other related works. Lastly, Section 7 concludes the paper.

2 DoS Attacks in SDWSNs

In this Section we describe in detail the three attacks investigated in this manuscript, their mode of operation, characteristics and the packets exchanged.

1. The **false flow request (FFR)** attack targets the controller and its main goals are to increase the controller's processing overhead and the packet traffic in order to increase the number of collisions. To attain these goals, the attacker sends multiple

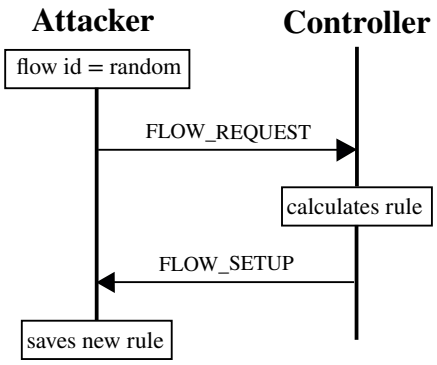


Figure 1: Flase flow request DoS attack

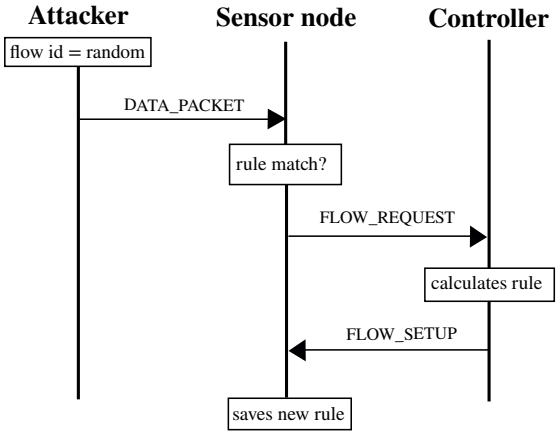


Figure 2: False data flow forwarding DoS attack

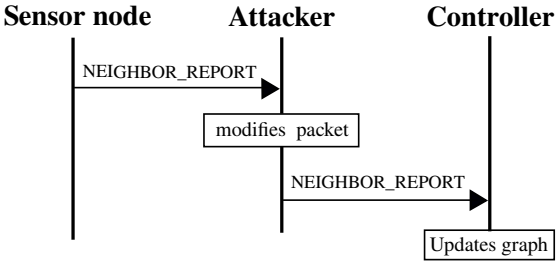


Figure 3: False neighbor information DoS attack

flow rule requests to the controller for different flows. The controller processes the packets, calculates the rules, and sends the replies to the attacker. Figure 1 depicts the packets' exchange during this attack.

2. The **false data flow forwarding (FDFF)** attack is similar to the FFR attack because it targets the controller, however its implementation is different as this is achieved via the network's devices. In this case the attacker sends a data packet with an unknown flow to its neighbors. Since the neighbors do not known the flow, they will ask a rule from the controller. From this point on, the attack behaves similarly to the FFR attack. The attack's main goals are to increase the controller's and neighbors' processing overhead and to increase the network's packet traffic. Additionally, this attack could saturate the neighbors' flow table (as these are constrained devices with limited storage). Figure 2 shows the packets' exchange for this attack.

3. The **false neighbor information (FNI)** attack modifies the packets that contain neighbor information. In this manner, the controller will mistreat false information as true and will send erroneous routing rules to the nodes. The main goal of this attack is to reduce the network's delivery rate. The packets' exchange for this attack is depicted in Figure 3.

Therefore, it becomes obvious that the attacks are of increasing aggressiveness, with an increasing number of packets being exchanged in each scenario. In particular, the FNI type of attack can cause significant performance degradation if not identified, as will be seen in the results Section.

3 IMPLEMENTATION

The three attacks were implemented using Contiki-OS version 3.0 [4] and IT-SDN version 0.4.1. [14]. Contiki-OS is a well known operating system for WSN and Internet of Things (IoT), and IT-SDN is an SDWSN framework and southbound protocol based on Contiki-OS. Next we explain the implementation of the the three attacks in detail.

3.1 False Flow Request

This attack was implemented using the function `sdn_send_data_flow_request(flowid_t f)` from IT-SDN. We use this function to send a flow rule request packet to the controller. The parameter f is the flow identifier for which the sensor node is requiring the rule.

In our implementation the attacker sends one flow rule request every T seconds and the parameter f is a random number between 0 and $65,535$. We chose this range because in IT-SDN the flow identifier is defined as an unsigned integer of 16 bits. The pseudocode for this attack is shown in Algorithm 1.

3.2 False Data Flow Forwarding

This attack requires that the attacker sends data packets to its neighbors using unknown flow numbers. To generate a packet with a flow number unknown to the controller, the attacker includes a new entry in his flow table with the unknown flow.

Algorithm 1: False Flow Request

```
start_timer(T)
while true do
    wait until new event
    if timer expired = TRUE then
        f = random(0, 65535)
        send_request(f)
        restart_timer
    end if
end while
```

Algorithm 2: False Data Flow Forwarding

```
start_timer(T)
while true do
    wait until new event
    if timer expired = TRUE then
        for n in neighbors do
            f = random(0, 1000)
            new_entry(neighbor_address,f, forward)
            send_data_packet(f)
        end for
        restart_timer
        erase_entries
    end if
end while
```

Algorithm 3: False Neighbor Information

```
while true do
    wait until new_packet to forward
    if new_packet is neighbor_report then
        modify_neighbors(new_packet)
        modify_metric(new_packet)
        forward(new_packet)
    end if
end while
```

Then, since the flow table has a limited memory allocation, the attacker erases this entry after sending the packet to avoid a flow table saturation. The IT-SDN function to include a new entry in the flow table has three parameters: the flow number, the next hop's address, and the action for the flow.

Similarly to our implementation of the FFR attack, in the FDFF attack scenario we use a timer to trigger an attack instance. When the timer expires, the malicious node creates n new entries in its flow table, where n is the number of its neighbors. Each entry has a different flow number and each number is determined using a random function. In this case the random function's range is between 10 and 1000, in order to minimize the probability of selecting randomly flow numbers that are actually being used in the network. The pseudocode for this attack is shown in Algorithm 2.

3.3 False Neighbor Information

In the FNI attack, the malicious node modifies the packet it sends to the controller, containing its neighbors' information. IT-SDN defines these packets as NEIGHBOR_REPORT packets. The information included in the NEIGHBOR_REPORT packets are the sensor node neighbors' addresses and the routing metric value for each neighbor.

When the attacker receives a packet for forwarding, it checks the packet type. In the case the packet is a NEIGHBOR_REPORT type, the attacker modifies the neighbors' addresses and the routing metric values. The pseudocode for this attack is shown in Algorithm 3.

4 METHODS

The main objective of this work is to understand the performance of an SDWSN when it is under different types of DoS attacks, captured in the present as FFR, FDFF, and FNI attacks. To attain our objective, we simulated each attack, varying the topology and the number of attackers. We use fully bidirectional square grid topologies from 36 to 121 nodes, and two aggressiveness levels: only one attacker and ten percent of nodes as attackers (rounded down to the closest integer value). We also simulated each scenario in the absence of attackers to use its performance results as a reference to determine the attacks' impact on the network performance. Each scenario was replicated ten times.

The simulations were performed using COOJA simulator [17]. COOJA allows to emulate different WSN platforms and simulate the radio medium. For the experiments we use sky mote, which is a TelosB mote [16] equivalent. The MAC layer is IEEE 802.15.4, configured to work without radio duty cycle (`nullrdc_driver`).

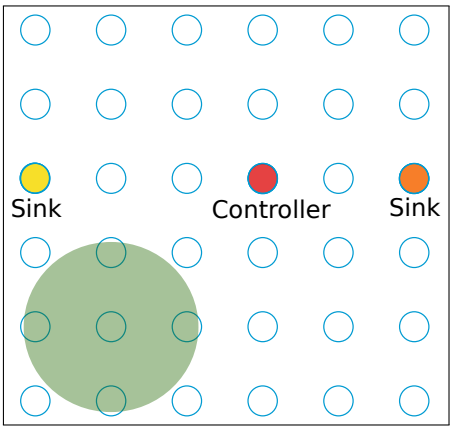
The topologies were configured with one data sink, one management sink, and one controller. The data sink receives the application's data, while the management sink receives data flow and control flow usage metrics. We use IT-SDN monitoring module [13] to implement the collection of flow usage metrics.

The controller and sinks positions in the network were placed according to the following rules:

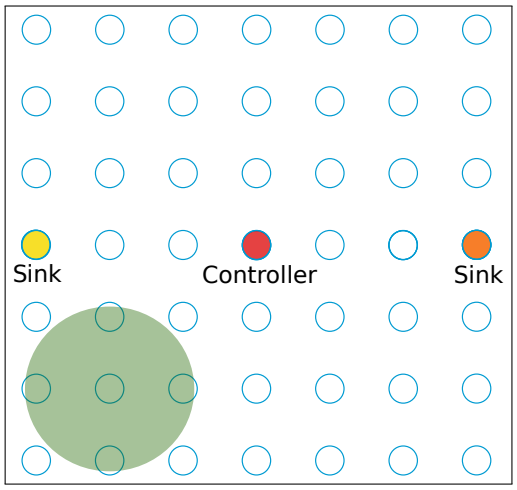
- The controller and the sinks are in the same row;
- For 49, 81, and 121 nodes, the controller is in the center of the grid, such as shown in Figure 4b.
- For 36, 64 and 100 nodes, the controller is in the lower left corner of the first quadrant (right-up), as shown in Figure 4a.

The sensor nodes transmit one data packet every minute and one management packet every 3 minutes. The data and management packets' payload is 10 bytes. In the case of the data packets, 10 bytes is enough to store simple measurement data. In the case of the management packets, we are monitoring two metrics: data flow and control flow usage. Such as explained in [13] two metrics is equivalent to 10 bytes.

IT-SDN 0.4.1 gives two options for the neighbor discovery protocol: the Contiki's Collect protocol [10] and a protocol for



(a) 36 nodes topology



(b) 49 nodes topology

Figure 4: Sink and controller position examples

directed IoT networks proposed by Alves *et al.* [3], devised to cope with unidirectional links. Since we are working with fully bidirectional networks, we decided to use the Collect protocol. Table 1 summarizes simulation, energy consumption and IT-SDN parameters.

Table 1: Simulation parameters

Simulation parameters	
Topology	Square grid
Number of nodes	36, 49, 64, 81, 100, 121
Simulation duration	3600 s
Node boot interval	$[0, 1]$ s
Number of sinks	2
Sinks position	Middle of the grid edge
Data traffic rate	1 packet per minute
Management traffic rate	1 packet every three minutes
Data payload size	10 bytes
Management payload size	10 bytes
Data traffic start time	$[2, 3]$ min
Radio module power	0 dB
Distance between neighbors	50 m
Attacks begins after	600 s

Energy Consumption parameters	
Transmission current consumption	21,70 mA
Receiving current consumption	22,00 mA
Processing current consumption	2,33 mA
Sleeping current consumption	0,18 mA
Operation voltage	3 V

IT-SDN parameters	
Controller position	Center
Controller retransmission timeout	60 s
ND protocol	Collect-based
Link metric	ETX
Neighbor report max frequency	1 packer per minute
CD protocol	None
Flow setup	Source routed
Route calculation algorithm	Dijkstra
Route recalculation threshold	20%
Flow setup types	Regular or source routed
Flow table size	10 entries

4.1 Attackers Configuration

The attackers were programmed to behave as regular nodes during the first 10 minutes of the simulations. After this time the attack is triggered. The FFR and the FDFF attacks work with a periodic timer that triggers the attacker's action. This timer was fixed to 60 seconds for both cases and for all scenarios. The FNI attack modifies the neighbors' addresses and the routing metric. IT-SDN uses two-byte addresses, thus the false addresses are generated randomly between 0 and $65,535$.

The routing metric is based on the expected transmissions (ETX) metric [6] calculated by the Collect protocol. The Contiki's Collect protocol implementation defines a minimum and maximum ETX of 8 and 511, respectively. The attacker uses this interval to randomly generate the false ETX value. The attackers' positions in the network were configured using the next rules:

- In the cases there is only one attacker, it is located in the third quadrant and is guaranteed to have 4 neighbors;
- In the case of multiple attackers, they have random positions but are equally distributed among the four quadrants in order to have the same number of attackers in each half of the grid and maximize the number of their non-malicious neighbors.

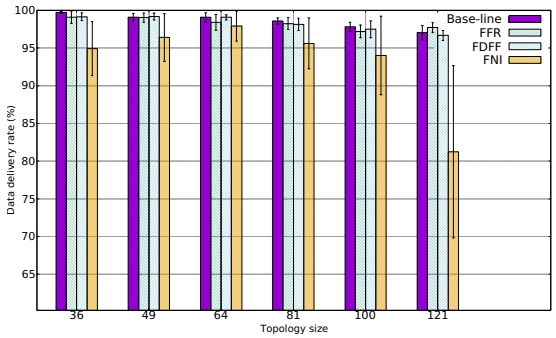


Figure 5: Data packet delivery rates for one attacker

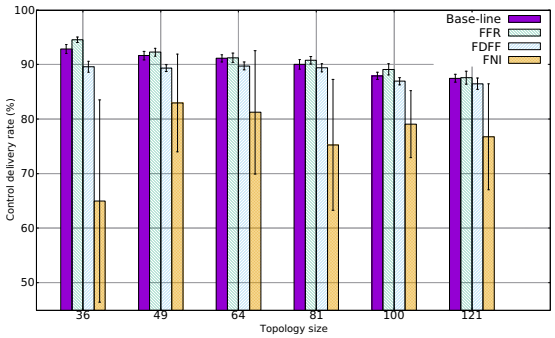


Figure 7: Control packets delivery for one attacker

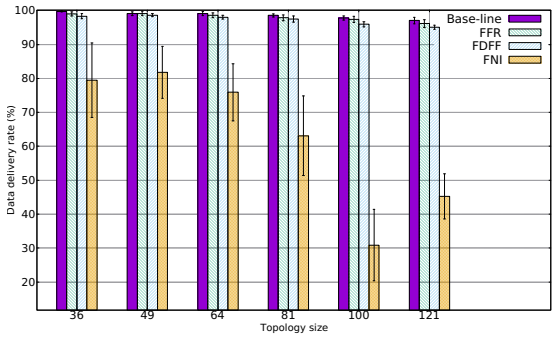


Figure 6: Data packet delivery rates for 10% of nodes as attackers

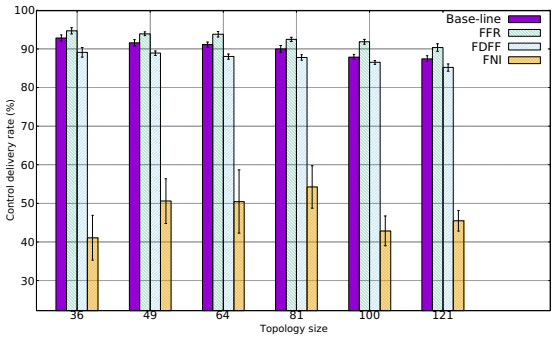


Figure 8: Control packets delivery for 10% of nodes as attackers

4.2 Performance Metrics

The metrics to measure the network performance are (i) the delivery rate, (ii) the end-to-end delay, (iii) the packet overhead, and, (iv) the energy consumption. The delivery rate is calculated dividing the total number of packets successfully received by the total number of packets sent. The end-to-end delay is the average time the packets spent to reach their destination. The overhead is quantified as the total amount of control packets per minute. Finally, the energy consumption is the average energy consumption of all nodes during one hour of simulation.

The delivery rate and the end-to-end delay were calculated for both control and data packets. The packet overhead was calculated only for control packets because the attacks were designed to increment specifically this type of traffic. The energy consumption was calculated for all nodes, excluding the sinks, the controller, and the attackers. We excluded the sinks and the controller because we assume those nodes have no energy restrictions and we excluded the attackers because we consider their energy consumption information could also be compromised.

5 Results and Discussion

In this section we present, analyze, and discuss the results obtained from our simulations. The results are sorted by metric and by attack aggressiveness.

Figure 5 depicts the data packet delivery rate when there is only one attacker in the network. These results show that when there is only one attacker in the network, the FFR and FDFF attacks do not have a significant impact on the delivery rate, unlike the FNI attack does. In all the scenarios when the network is under a FNI attack, the average delivery rate is lower than the delivery rate in the baseline and the other attacks. Furthermore, we note a considerably higher variability (shown here as the standard deviation of the measurements) in the data packet delivery rates when compared to the baseline reference performance.

Figure 6 depicts the data packet delivery rate when ten percent of nodes are malicious. The results show that the networks under the FFR attack and the networks from 36 to 81 nodes under the FDFF attack maintain the baseline delivery rate. The networks with 100 and 121 nodes under the FDFF attack and all the network scenarios under the FDN attack show a decrease in the delivery rate with respect to the baseline results. In the case of the FDFF attack, the delivery rate drop is less than 2%, while in the case of the FNI attack the drop is considerably higher, between 17% and 66%.

The delivery rate results for control packets are shown in Figure 7 and Figure 8 for one attacker and for multiple attackers, respectively. Similar to the previous set of results for the data packet delivery rates, the FNI is the attack that has the highest impact on the delivery rate. On the other hand, we observed two behaviors that were not present in the data packets results: (i) all the networks under the FFR attack with

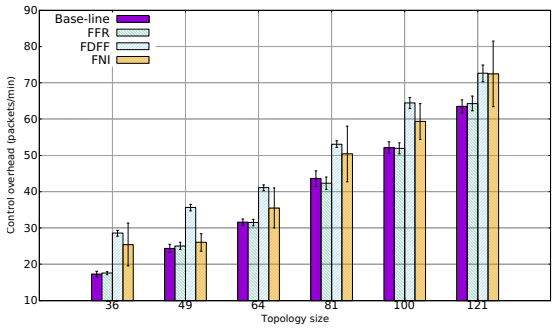


Figure 9: Control packets overhead for one attacker

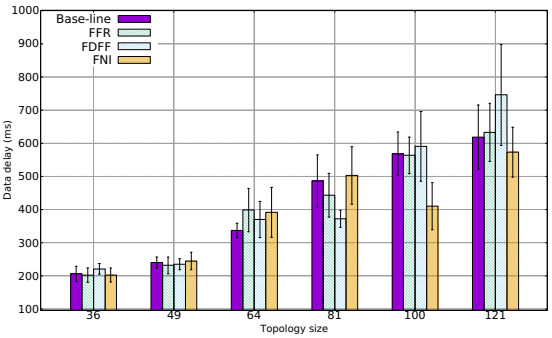


Figure 12: Data packets delay for one attacker

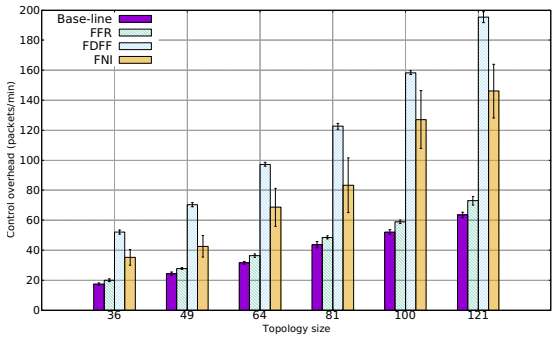


Figure 10: Control packets overhead for 10% of nodes as attackers

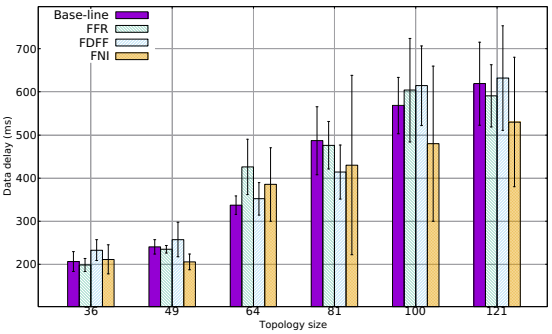


Figure 13: Data packets delay for 10% of nodes as attackers

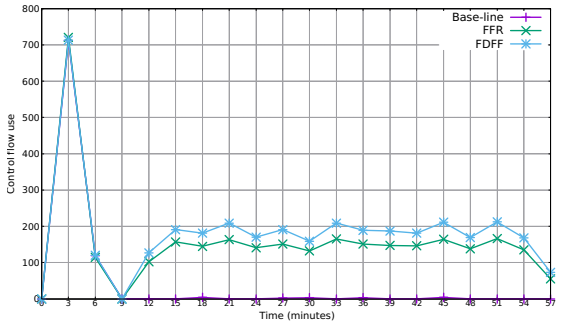


Figure 11: Control packets flow usage

ten percent of nodes as attackers showed an improvement in the delivery rate for the control packets; and (ii) all the networks under the FDFF attack with multiple attackers showed a drop in their control packets delivery rate.

To understand both behaviors, it is necessary to analyze also the total control packets overhead and the control packets traffic during the simulation time. The results in Figure 9 and Figure 10 show that the networks under the FDFF attack have the largest control packets overhead for both single and multiple attackers. The FFR attack does not impact the control packets overhead when there is only one attacker, but induces an increase in this metric when there are multiple attackers. Then, Figure 11 shows a large control packet traffic due to the network configuration during the first three minutes, and then this number falls to zero. Subsequently, after the initial exchange of a large number of control packets the first three minutes of operation, in the baseline scenario the number of control packets continues to be very small, close to zero. On the other hand, in the FFR and in the FDFF attacks the control packets flow usage increases over 100 times once the attacks are launched.

Therefore, the control packets delivery rate obtained during the baseline scenario concerns mostly the first three minutes. On the other hand, when the attackers start to operate, there is constant but less dense control traffic than in the first three minutes. This is why the control packets metric has a larger value in the FFR attack scenario compared to the baseline. The situation is different for the FDFF attack mainly for two reasons: (i) the control packets overhead is larger than the control packets overhead in the FFR attack, and (ii) all the attacker's neighbors send a flow rule request at the same time, instead of only the single attacker sending one flow rule request per minute. Both situations increase the probabilities of collisions.

The control packets overhead results also give information about the FNI attack. When there is only one attacker the average overhead in all the topologies sizes increases compared to the baseline results; interestingly, we also see an increase in the standard deviation of this metric. In the case of multiple attackers, there is a high increase in the control packets overhead, with the corresponding gap – when compared to the baseline results – increasing with the number of nodes. For

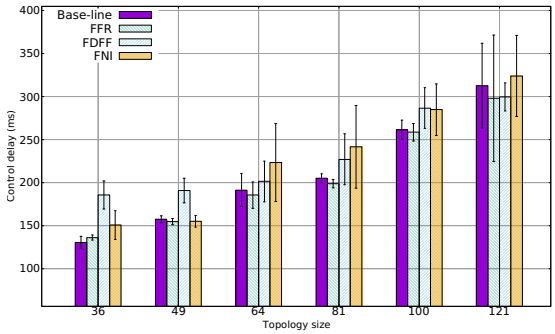


Figure 14: Control packets delay for one attacker

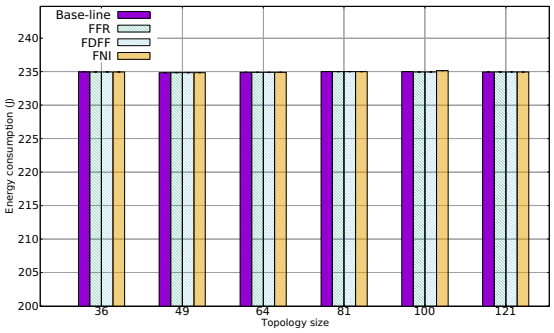


Figure 16: Energy consumption for one attacker

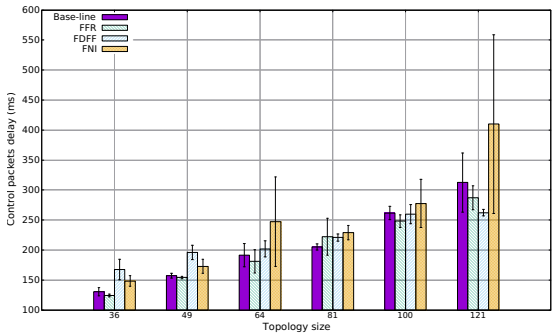


Figure 15: Control packets delay for 10% of nodes as attackers

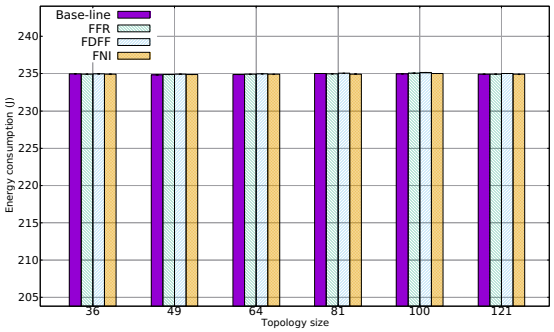


Figure 17: Energy consumption for 10% of nodes as attackers

36 nodes, the average control packets per minute increases by $102,78\%$, for 64 nodes it increases by $117,52\%$, and for 121 nodes it increases by $130,07\%$.

The data packets delay results depicted in Figure 12 and Figure 13 show a high dispersion for topologies from 64 to 121 nodes. Also, this dispersion is higher when there are multiple attackers in the network. Additionally, the mean values do not show any pattern along the different topologies sizes that could aid to determine the impact of each attack in the data packets delay. On the other hand, our baseline delay results coincide with the results obtained in [14] and [3] for topologies over 49 nodes. This indicates that it is common to have high dispersion in this metric even though the network is not under attack.

In the case of the control packets delay results shown in Figure 14 and Figure 15, the networks with 36 and 49 nodes under the FDFF attack have the highest delay. This behavior is the same when there is only one attacker and when there are multiple attackers. For topologies over 49 nodes, the dispersion in the metrics increases and the difference among the networks under attack delay results becomes less clear. For this reason, the dispersion in the delay metric render it difficult to determine the impact of each attack in the control packets delay when monitoring the average delay only.

Finally, the energy consumption results are shown in Figure 16 and Figure 17 for one attacker and multiple attackers, respectively. In both cases, the energy consumption remains unchanged, which means the attacks do not induce an energy consumption overhead for the network. On the other hand, all the nodes in the network were programmed to work without radio duty cycle, which means the radio module is turned on all the time. Since the radio module has the highest energy consumption in the node, the energy consumption overhead generated by the attacks becomes negligible. In the future, we will run experiments with radio duty cycle turned on to obtain a finer understanding of the attacks' impact on the energy consumption.

Summarizing, the FFR attack does not induce a significant change in the network delivery rate, packets overhead and end-to-end delay. This means that those metrics are not the proper indicators to detect this type of attack; at the same time, this attack is very mild and does not heavily impact the performance of the network. On the other hand, the control packets flow usage analysis offered useful information about the difference with the baseline scenario in terms of control packets traffic. We have identified a small increase in the average number of control packet delivery both for single and multiple attackers, making this metric a potential candidate for the identification of this type of attacks.

The scenarios under the FDFF attack indicated the highest control packets overhead and also a drop in the control packets delivery rate. Conversely, this attack did not alter the metrics related to the data plane. As a result, it is conceivable that a joint monitoring of the control packet overhead and of the control packet delivery rate can offer the means to identify this type of attacks.

The FNI attack was the only one that affected both control

Table 2: Related work comparison

Work	Approach	Attacks	Implementation/ Simulation	Parameters	Metrics
Anhtuan et al. [12]	IoT	Rank	Contiki 2.5/COOJA	attack aggressivenes	delivery rate, delay
Tripathi et al. [22]	WSN	Black Hole, Gray Hole	NS-2	topology size (20-200)	lifetime, energy consumption, delivery rate
Alanazi et al. [2]	IoT	Hello flooding	Not-specified	attack aggressivenes	delivery rate, throughput, and delay
ETMRM [23]	SDWSN	Selective forwarding New Flow	SDN-WISE/COOJA	attack aggressiveness	control overhead, network lifetime, energy consumption, data packets loss ratio
This work	SDWSN	False Flow Request, False Data Forwarding, False Neighbor Information	IT-SDN/COOJA	attack aggressiveness topology size (36-121)	delivery rate, delay, packets overhead

and data packets metrics, both in terms of the average value of the metrics as well as of their dispersion. This attack reduced the control and data packets delivery rate and increased the control packets overhead. Also, the performance results show there is a significant difference when there is only one attacker and when there are multiple attackers. We posit that monitoring the variability of these metrics might offer the means to identify this type of attacks.

Next, when looking at the energy consumption, our results showed that when the duty cycle is off, the attacks do not affect the average network's energy consumption. Lastly, the end-to-end delay metric used in this work did not give results that, at present, we could consider bearing any useful information in identifying the attacks based on their impact on the network performance. The main problem was the high dispersion in the results, which is consistent with previously published work. In the future, we will explore whether an approach to separate the delay metric by number of hops to reach the controller and the sink, or by clusters, could be an option to reduce the dispersion.

6 Related Work

Traffic and performance analysis is a technique commonly used in WSN and IoT to detect malicious nodes in the network. In this section we briefly review previous works that have studied the impact of security attacks in the network performance and highlight the contribution of this work.

Anhtuan et al. [12] study the impact of Rank attacks on networks with RPL (IPv6 Routing Protocol for Low-Power and Lossy Networks - RFC 6550). The impact in the network performance was measured through the average end-to-end delay, delivery rate, number of affected nodes and number of DIOs generated (RFC 6550). The experiments were conducted using Contiki 2.5 and COOJA simulator, using one grid of 100 nodes, multiple attackers, and simulations of 350 seconds.

Black Hole and Gray Hole attacks are common in WSN and IoT. Both attacks are devised to collect and drop packets, impeding or delaying these packets to reach the sink. Tripathi *et al.* [22] compare the impact of both attacks in a WSN with the LEACH routing protocol [9]. The performance metrics used in this work are: network lifetime, delivery rate, and energy consumption. Simulations were conducted using NS-2, varying the network size from 20 to 200 nodes.

The impact of a "Hello Flooding" attack in a IoT-based network is presented in [2]. The Hello Flooding attack was simulated in a pseudo-random topology with 30 nodes, using the AODV routing protocol and varying the number of attackers from one to four. Also, authors test the resilience of the network when using PASER [18], a protocol to combat unauthorized nodes of joining the network. The metrics to measure the impact on the network performance were: delivery rate, throughput, and delay. Authors claim PASER is a good candidate to secure IoT networks.

ETMRM [23] is a routing and management mechanism for SDWSN devised to handle malicious forwarding attacks. This work tested the performance of ETMRM using two attacks: Selective Forwarding attack, also known as Gray Hole attack, and New-flow attack [24]. The implementation was conducted using SDN-WISE [5]. The experiments were conducted using COOJA [17], simulating a network with 100 nodes during 300 seconds. The nodes were deployed in a random positions and varying the number of attackers. The network performance metrics used are: control overhead, network lifetime, energy consumption, and data packets loss ratio.

From the papers reviewed before, we noticed a scarcity of works studying the impact of SDWSN specific attacks on the network performance. There is a lack of attack aggressiveness experiments on different topology sizes. To fill this gap in the literature, we implemented three different SDWSN attacks and tested them on six topology sizes. We measured the impact of each attack on each topology varying the attack aggressiveness. Table 2 summarizes the related work review.

7 Conclusions

Software-defined networking has been identified as a solution for many WSN challenges concerning flexibility and resource reuse. On the other hand, the SDN architecture is exposed to new security threats.

In this work we implemented three security attacks

showcasing potential SDWSN vulnerabilities and analyzed their impact on the network performance considering four metrics: the delivery rate, the packet overhead, the end-to-end delay, and, the energy consumption. Our results show that the first of the attacks studied, the False Flow Request attack, did not have a significant impact on any of the performance metrics used. On the other hand, the results obtained for the False Data Flow Forwarding and the False Neighbor Information attacks showed significant changes in the control and data packets delivery rate, and in the control packets overhead.

The energy consumption results showed the attacks did not incur a significant energy consumption overhead, while the end-to-end delay results were inconclusive due to the high dispersion in all the scenarios. To use a delay metric based on the number of hops to reach the controller and the sink, or by clusters, could be an option to reduce the dispersion.

In future work, we will use more than two levels of aggressiveness, for example, increasing the number of packets per minute the attacker sends to the controller. We will also test the relation between the network performance with the attacker position, and the impact on the energy consumption when using a radio duty cycle mechanism. Our ultimate goal is to develop lightweight algorithms for the early detection of these and similar DoS types of attack; as an example, monitoring the average value of the control packet delivery rate and overhead emerge as potential candidates for identifying FFR and FDFF attacks, while monitoring changes in the variance of these metrics could be better suited to identify FNI type of attacks.

ACKNOWLEDGEMENTS

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001 and by FAPESP (São Paulo Research Foundation), process 2018/12579-7. Gustavo A. Nunez Segura is supported by Universidad de Costa Rica. A. Chorti is supported by the ELIOT ANR-18-CE40-0030 project.

REFERENCES

[1] G. A. Ajaeiya, N. Adalian, I. H. Elhajj, A. Kayssi, and A. Chehab, "Flow-based intrusion detection system for sdn," in *2017 IEEE Symposium on Computers and Communications (ISCC)*, July 2017, pp. 787–793.

[2] S. Alanazi, J. Al-Muhtadi, A. Derhab, K. Saleem, A. N. AlRomi, H. S. Alholaibah, and J. J. P. C. Rodrigues, "On resilience of wireless mesh routing protocol against dos attacks in iot-based ambient assisted living applications," in *17th International Conference on E-health Networking, Application Services (HealthCom)*, Oct 2015, pp. 205–210.

[3] R. C. A. Alves, C. B. Margi, and F. A. Kuipers, "No way back? An SDN protocol for directed IoT networks," in *15th Wireless On-demand Network systems and Services Conference*. IEEE, Jan. 2019.

[4] A. Dunkels, B. Gronvall, and T. Voigt, "Contiki - a lightweight and flexible operating system for tiny networked sensors," in *29th Annual IEEE International Conference on Local Computer Networks*, 2004, pp. 455–462.

[5] L. Galluccio, S. Milardo, G. Morabito, and S. Palazzo, "SDN-WISE: Design, prototyping and experimentation of a stateful SDN solution for WIreless SEnsor networks," *INFOCOM*, vol. 26, pp. 513–521, 2015.

[6] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P. Levis, "Collection tree protocol," in *Proceedings of the 7th ACM conference on embedded networked sensor systems*. ACM, 2009, pp. 1–14.

[7] T. Ha, S. Kim, N. An, J. Narantuya, C. Jeong, J. Kim, and H. Lim, "Suspicious traffic sampling for intrusion detection in software-defined networks," *Computer Networks*, vol. 109, pp. 172 – 182, 2016.

[8] E. Haleplidis, K. Pentikousis, S. Denazis, J. H. Salim, D. Meyer, and O. Koufopavlou, "Software-Defined Networking (SDN): Layers and Architecture Terminology," IETF, RFC 7426, Jan. 2015. [Online]. Available: http://tools.ietf.org/rfc/rfc7426.txt

[9] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Transactions on Wireless Communications*, vol. 1, no. 4, pp. 660–670, Oct 2002.

[10] J. Ko, J. Eriksson, N. Tsiftes, S. Dawson-Haggerty, J.-P. Vasseur, M. Durvy, A. Terzis, A. Dunkels, and D. Culler, "Industry: Beyond interoperability: Pushing the performance of sensor network ip stacks," in *Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems*, ser. SenSys '11, New York, NY, USA, 2011, pp. 1–11.

[11] H. I. Kobo, A. M. Abu-Mahfouz, and G. P. Hancke, "A Survey on Software-Defined Wireless Sensor Networks: Challenges and Design Requirements," *IEEE Access*, vol. 5, pp. 1872–1899, 2017.

[12] A. Le, J. Loo, A. Lasebae, A. Vinel, Y. Chen, and M. Chai, "The impact of rank attack on network topology of routing protocol for low-power and lossy networks," *IEEE Sensors Journal*, vol. 13, no. 10, pp. 3685–3692, 2013.

[13] T. C. Luz, G. A. Nunez, C. B. Margi, and F. L. Verdi, "In-network performance measurements for Software Defined Wireless Sensor Networks," in *2019 IEEE 16th International Conference on Networking, Sensing and Control (ICNSC)*, May 2019, pp. 206–211.

[14] C. B. Margi, R. C. A. Alves, G. A. N. Segura, and D. A. G. Oliveira, "Software-Defined Wireless Sensor Networks Approach: Southbound Protocol and Its Performance Evaluation," *Open Journal of Internet Of Things (OJIOT)*, vol. 4, no. 1, pp. 99–108, 2018, special Issue: Proceedings of the International Workshop on Very Large Internet of Things (VLIoT 2018) in conjunction with the VLDB 2018 Conference in Rio de Janeiro, Brazil. [Online]. Available: http://nbn-resolving.de/urn:nbn:de:101:1-201808051930571018960 7

[15] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner,

"OpenFlow: enabling innovation in campus networks," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 2, p. 69, 2008.

[16] MEMSIC Inc., *TelosB datasheet: Document Part Number: 6020-0094-02 Rev B*, MEMSIC Inc., San Jose, California, 2003.

[17] F. Österlind, A. Dunkels, J. Eriksson, N. Finne, and T. Voigt, "Cross-level sensor network simulation with COOJA," in *Proceedings of Conference on Local Computer Networks*, 2006, pp. 641–648.

[18] M. Sbeiti, N. Goddemeier, D. Behnke, and C. Wietfeld, "PASER: Secure and Efficient Routing Approach for Airborne Mesh Networks," *IEEE Transactions on Wireless Communications*, vol. 15, no. 3, pp. 1950–1964, March 2016.

[19] P. Sinha, V. K. Jha, A. K. Rai, and B. Bhushan, "Security vulnerabilities, attacks and countermeasures in wireless sensor networks at various layers of osi reference model: A survey," in *2017 International Conference on Signal Processing and Communication (ICSPC)*, July 2017, pp. 288–293.

[20] S. Skaperas, L. Mamatas, and A. Chorti, "Early Video Content Popularity Detection with Change Point Analysis," in *IEEE Global Communications Conference (GLOBECOM)*, Abhu-Dhabi, United Arab Emirates, Dec. 2018. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01997965

[21] I. Tomic and J. A. McCann, "A Survey of Potential Security Issues in Existing Wireless Sensor Network Protocols," *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 1910–1923, Dec 2017.

[22] M. Tripathi, M. Gaur, and V. Laxmi, "Comparing the Impact of Black Hole and Gray Hole Attack on LEACH in WSN," *Procedia Computer Science*, vol. 19, pp. 1101 – 1107, 2013.

[23] R. Wang, Z. Zhang, Z. Zhang, and Z. Jia, "Etmrm: An energy-efficient trust management and routing mechanism for sdwsns," *Computer Networks*, vol. 139, pp. 119 – 135, 2018.

[24] T. Xu, D. Gao, P. Dong, H. Zhang, C. H. Foh, and H. Chao, "Defending Against New-Flow Attack in SDN-Based Internet of Things," *IEEE Access*, vol. 5, pp. 3431–3443, 2017.

AUTHOR BIOGRAPHIES

Gustavo A. Nunez Segura is a PhD student at Universidade de São Paulo. He received the M.Sc. degree (2018) in Electrical Engineering from Universidade de São Paulo and the B.Sc. in Electrical Engineering from Universidad de Costa Rica. His main research interests include energy consumption and security in wireless sensor networks and software-defined networking.

Cintia Borges Margi obtained her Ph.D. in Computer Engineering at University of California Santa Cruz (2006) , and her Habilitation (Livre Docencia) (2015) in Computer Networks from the University of Sao Paulo. She is Associate Professor in the Computer and Digital Systems Engineering department at Escola Politecnica – Universidade de São Paulo (EPUSP) since 2015, where she started as Assistant Professor in 2010. During 2007-2010 she was Assistant Professor at Escola de Artes, Ciencias e Humanidades da Universidade de São Paulo (EACH-USP). Her research interests include: wireless sensor networks and software-defined networking.

Arsenia Chorti is an Associate Professor (MCF) at ENSEA since Sept. 2017. She obtained her PhD from Imperial College and from 2010 to 2012 was as a Marie Curie Int. Outgoing Fellow (MC-IOF) at Princeton University where she is currently a visiting researcher. She served as Lecturer and Senior Lecturer at the Universities of Middlessex and Essex between 2008-2017. Her research interests include, PLS, 5G, resource allocation. She is a member of the IEEE Teaching Awards Committee, Associate Editor of the Springer Internet Tech. Letters and of Springer Emerging Telecommunications. She participated in many EU, EPSRC and CNRS research projects.