On Mitigating DIS Attacks in IoT Networks

Ghada Aljufair[1], Mohammed Mahyoub [2§], and Abdulaziz S. Almazyad[3]

[1,3]Department of Computer Engineering, KSU, Saudi Arabia
[2]School of Information Technology, Carleton University, Canada
[1]aljufairghada@hotmail.com, [2]mohammed.mahyoub@carleton.ca, [3]Mazyad@ksu.edu.sa

***Abstract*—Routing protocols deem a pivotal component of the communication stack in the Internet of Things (IoT). The ipv6 Routing Protocol for Low power and lossy networks (RPL) has been standardized by the Internet Engineering Task Force (IETF) for routing in IoT-based networks. RPL-related control messages are transmitted in the network to construct an optimized forwarding structure. A malicious insider node can attack RPL networks by sending a high number of unnecessary control messages which causes a detrimental side effect on the network performance. One of these attacks targets DIS control messages transmitted by a new node to join the network. This attack is called the DIS attack. The attacker can exploit the joining process to flood the network with a large volume of DIS messages. This paper aims to investigate the effect of DIS attacks on network performance and develop an effective technique to mitigate the adverse effects of such attacks. The proposed technique is implemented in the Contiki operating system and evaluated using the Cooja emulator. Compared to the standard RPL and other comparable work in the literature, the proposed technique retains low routing control cost, high throughput, and low energy consumption.**

***Index Terms*—IoT, LLNs, RPL, Security, DIS Attacks, Contiki.**

I. Introduction

The Internet of Things (IoT) was defined by the International Telecommunication Union (ITU) as "a global infrastructure for the information society, enabling advanced services by interconnecting physical and virtual things based on existing and evolving interoperable information and communication technologies" [1]. The high interest in IoT applications resulted in the large-scale deployment of Low power and Lossy Networks (LLNs) which represent the main building blocks of IoT [2]. LLNs are widely used in many applications such as industrial monitoring, home automation, environmental monitoring, urban sensing networks, healthcare, energy management, asset tracking, and cooling systems [3].

Traditionally, LLNs are composed of nodes with limited resources in terms of processing power, memory capacity, bandwidth, and battery-based power. Moreover, they have high faulty links, high loss rates, and low data rates. Due to these limitations, traditional Internet routing protocols are unsuitable and they need to be adapted in some ways. For this, the IPv6 **R**outing **P**rotocol for **L**LNs (**RPL**) has been standardized to satisfy the routing requirements of such networks [4]. RPL organizes the LLN nodes in a tree-like structure typically rooted at the border router known as a destination-oriented directed acyclic graph (DODAG).

Having said that, RPL provides little security against different routing attacks [5]. These attacks make hampered enforce basic security services like confidentiality, data integrity, authenticity, and access control. Attacking restricted resources in RPL-based networks would affect the overall network performance. In a DODAG Information Solicitation (DIS) attack, for example, a malicious node sends DIS messages periodically to its neighbors to solicit DODAG Information Object (DIO) messages [6]. The attacker forces legitimate nodes to respond with many unnecessary DIO packets which consume their already limited resources and significantly affect the network performance. In this paper, we first conduct an effect and feasibility study on the impact of the DIS attack against RPL-based networks. Second, we develop a mitigation technique against such attacks. The proposed technique, named DIO response-based mitigation, is implemented in the Contiki operating system and evaluated using the Cooja emulator in terms of the routing traffic overhead, energy consumption, and packet delivery ratio (i.e. throughput).

The rest of this paper is structured as follows. Section II provides a brief overview of the RPL protocol and DIS attacks. Section III reviews the related work. Section IV discusses the proposed mitigation technique against DIS attacks. The feasibility and the effectiveness of the proposed mitigation technique are presented in Section V. Finally, Section VI concludes the paper.

II. RPL and DIS Attacks Overview

In this section, we give an overview of the RPL, discuss how the routing structure (i.e. the DODAG) is constructed, and explain the DIS attacks in some detail.

A. RPL Overview

RPL protocol is a proactive distance-vector routing protocol based on IPv6. Three messages of Internet control message protocol (ICMP) type are used in the DODAG construction process. These messages are DIO, DIS, and Destination Advertisement Object (DAO) [7]. The DODAG Root (i.e. the border router) initiates the DODAG construction process by broadcasting a DIO message to all surrounding neighbors. The DIO message contains the routing information, rank value, and other configuration parameters of the existing DODAG needed for the joining process. Upon receiving the DIO messages,

§The author participated in this work during his PhD study at KFUPM

 ISBN: 978-3-903176-56-0

a joining node calculates its rank based on the predefined objective function (OF), selects its preferred parent, and finally updates the received DIO with its rank and re-broadcasts it further. This process is continued until all participating nodes are covered. At the end of this process, the upward routes from nodes toward the DODAG root are built [8].

The dissemination of DIO messages is controlled by the Trickle Algorithm (TA) [9]. It decides when a node can send its subsequent DIO messages to maintain consistency in a network. If the node receives consistent messages (i.e. a DIO message without any change compared to the one it plans to send), the next sending interval is doubled. This means that the network is stable and no more changes in the DODAG structure occur in the current interval. On the other hand, if the node receives inconsistent messages, it resets its next interval to the minimum length. As a result, when the network is stable the time interval increases leading to fewer DIO messages transmitted. However, when the network is unstable, the trickle timer is reset to send DIO messages more frequently [9].

To build the downward routing, the joining node unicasts its routing information to its preferred parent. This information is carried out in a DAO message. In fact, DAO messages are unicasted further until they reach the DODAG root. The received DAO message is handled based on the mode of operation used. There are two modes defined by the RPL specification, storing and non-storing. The used mode is advertised in DIO messages. The node receiving a DAO message may respond back by DAO acknowledgment. For point-to-point (i.e. node-to-node) routing, packets are forwarded based on the routing mode used in the network. In the case of the non-storing mode, they are either routed all the way up to the root node and then back to the destination node, or up to a common ancestor node and then down to the destination in the case of the storing mode. [10].

Finally, when a new node wants to join the network, it waits for a predefined time to receive DIO messages to be used for the joining process. If it does not receive a DIO, the node sends a DIS message to its neighbors to solicit a DIO containing the network routing information. On the other hand, if the node receives a DIO message, it stops sending DIS messages and sends a DAO message back to the sender node to join the network.

B. DIS Attacks Overview

This subsection explains the DIS attack and how victim nodes behave under this attack. As mentioned earlier, when a normal node wants to join, it waits for a predefined time to receive a DIO message from its neighbors. If no DIO is received, it sends a unicast or multicast DIS message to solicit a DIO. This event is repeatedly done on a fixed basis until a DIO message is received. Upon receiving a unicast DIS message, the receiving node responds with a DIO message to the sender node without resetting its trickle timer. If a multicast DIS message is received, the receiving node resets its trickle time and responds with a multicast DIO message. On the other hand, if the joining node behaves maliciously, it will continue to transmit multicast DIS messages periodically to its neighbors even if it is already received a DIO back. Sending DIS messages periodically forces receiving nodes to reset their trickle timer to respond by a multicast DIO.

Algorithm 1 shows DIS attacks. In the algorithm, n, $List_{Attack}$, T_d, and DIS_{intr} refer to a node, a list of the attacking nodes, the time delay before the transmission of the first DIS, and the DIS interval, respectively. After a node starts up and T_d expires, it checks if its ID, n_{id}, is in $List_{Attack}$. If it does, it then sends DIS periodically at the DIS_{intr} or it behaves normally otherwise.

Algorithm 1 DIS Attack Algorithm

Input: N, n, $List_{Att}$, T_d DIS_{intr}
Output: DIS

```
1:  Boot-up
2:  for all n ∈ N do
3:      Set T_d
4:      if T_d expires then
5:          if n_id ∈ List_Att then
6:              while true do
7:                  Send DIS
8:                  Wait One Second
9:              end while
10:         else
11:             if no DIO received and DIS_intr is reached then
12:                 Send DIS
13:             else
14:                 Suppress DIS
15:             end if
16:         end if
17:     end if
18: end for=0
```

For example, as shown in Fig. 1, the malicious node of ID# 9 regularly sends multicast DIS messages to its neighbors of IDs# 6,7, and 8. This triggers those nodes to reset their trickle timer to I_{min} and respond by multicasting DIO messages. Resetting the trickle timer repeatedly upon receiving multicast DIS messages increases control packet overhead, which in turn increases the power consumption of non-malicious nodes (i.e. victim nodes) accordingly. Moreover, the performance of the network is significantly degraded.

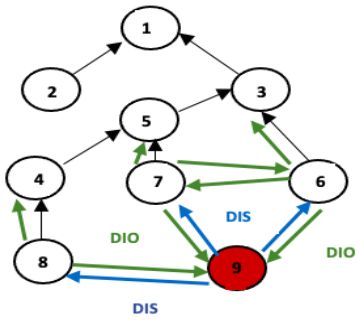

Fig. 1. Example of DIS Attack showing node ID# 9 sends DIS messages which enforce its neighbors of IDs# 6, 7, and 8 to respond by multicast DIOs

III. Related Work

This section reviews the literature studies that proposed mitigation techniques against RPL-based DIS attacks.

The most notable related work was presented by Verma *et al.* in [11]. They proposed a lightweight mechanism to mitigate DIS attacks named Secure-RPL. The main idea of their solution is to restrict the number of DIOs that are sent back in response to DISs received from the neighborhood. This is achieved by making each node track the number of DISs received from its neighbors. Upon receiving a DIS from a neighbor node, the receiving node checks the total number of DISs corresponding to the sending node. The receiving node then resets the trickle timer if and only if the checked number is less than a predefined threshold. This helps to avoid unnecessary resets of the trickle timers, which in turn reduces control message transmissions triggered by the DIS attack.

Farzane *et al.* in [12] presented an anomaly-based intrusion detection system that is based on threshold values to detect DIS attacks. Their idea is as follows, each node calculates the number of DIS messages received from each neighbor. If the number of DIS messages exceeds a predefined threshold in a particular interval, the receiving node blocks the sending neighbor by discarding its request to solicit a DIO. Their results demonstrated that the overhead is decreased using the proposed method, however, the packet delivery ratio was not evaluated.

Guo in [13] proposed a lightweight countermeasure against DIS attacks. The idea of their proposed scheme is that every node monitors its neighbors using the intervals between two consecutive DIS messages and its total count to a threshold value. Two thresholds are introduced for detecting malicious behaviors. One is for measuring the interval between consecutive DIS messages, and the other one is for monitoring the total number of these messages.

In summary, all literature studies demonstrate the same concept for mitigating DIS attacks. In order to suppress the DIOs sent back in response to DIS messages, they use a predefined threshold for the number of DIS messages received by a particular node or all neighboring nodes.

IV. The Proposed DIO Response-based Mitigation Technique

Our proposed approach to mitigate DIS attacks is called DIO response-based mitigation DIO_{resp} and is discussed in this section. It is important to note that the proposed technique in this study is built on the solution of Verma *et al.* [11] discussed in the previous section and referred to herein as DIS_{thr} for the sake of comparison purposes. The main idea of our proposed approach DIO_{resp} is to restrict the number of transmitted DIOs by adding a new condition to send a DIO in the next trickle interval. The new condition considers the total number of DIOs received from neighbors, particularly those sent in response to DIS messages. This number is tracked by all neighboring nodes. If this number exceeds a predefined threshold, a node suppresses its DIO transmission. To know whether a received DIO is a response to a DIS message, the receiving node checks a flag on the DIO message, which has already been set by the DIO sender. This flag is set in the DIO object structure intended to be sent as a response. From the implementation viewpoint, one of the reserved fields in the DIO format is used as a flag. Figure 2 shows the structure of the DIO message, including the flag used.

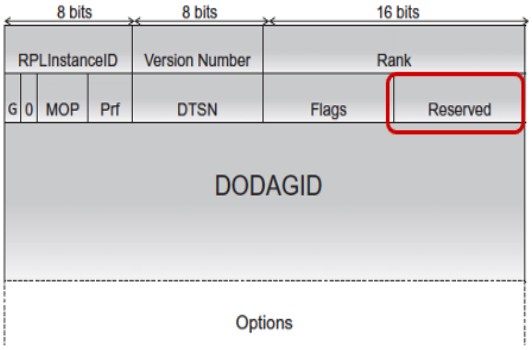

Fig. 2. The base object format of the DIO message

For more clarification, algorithm 1 shows the proposed solution. In this algorithm, N, DIO_{flg}^{resp}, DIS_{thr}, and T_{trkl} refer to the list of all nodes in the network under study, the flag of the DIO response indicator, the predefined DIS threshold, and the trickle timer, respectively. Additionally, T_{dio}^{snd}, K, and DIO_{thr}^{resp} refer to the selected time in which the DIO is supposed to be sent, the TA's redundancy factor, and the DIO response threshold, respectively. Finally, DIS_{cnt}, DIS_{tot}, C, and DIO_{tot}^{resp} refer to the number of DIS messages received from a particular node, the total number of DIS messages received from a particular node, the number of DIO messages received from all neighbors, and the number of total DIO response messages received from all neighbors, respectively.

Algorithm 1 DIO_{resp} Algorithm

Input: N, DIS, DIO_{flg}^{resp} T_{dio}^{snd}, K and DIO_{thr}^{resp}
Output: $DIOScheduling$

```
for all n ∈ N do
    Initialize DIS_cnt, DIS_tot, C and DIO_tot^resp
    if DIS received then
        DIS_cnt ++
        if DIS_tot <= DIS_thr then
            Reset T_trkl
            Set DIO_flg^resp
        end if
    end if
    if DIO received then
        if DIO_flg^resp == 1 then
            DIO_tot^resp ++
        end if
    end if
    if T_dio^snd expires then
        if C < K and DIO_tot^resp <= DIO_thr^resp then
            Send DIO
        else
            Suppress DIO
        end if
    end if
end for=0
```

Each node keeps tracking the number of DIS messages received from each neighboring node and the number of DIO responses from all neighbors as well. When a node receives a multicast DIS message, it increments the corresponding DIS_{cnt} of the sender node if the DIS_{tot} is higher than the DIS_{thr}, the T_{trkl} will not be reset. Otherwise, the T_{trkl} is reset, and the DIO_{flg}^{resp} is set. This flag is added to the DIO sent at the subsequent trickle interval. Upon receiving a DIO message, the receiving node checks the DIO_{flg}^{resp}, and if this flag was set, the DIO_{tot}^{resp} is incremented. A sending node sends a DIO if and only if the DIO_{tot}^{resp} counter is less than DIO_{thr}^{resp}. This condition is added to the condition already in place related to the C counter and redundancy factor K of the TA.

V. Results and Discussion

This section shows the feasibility and effectiveness of the proposed mitigation technique, DIO_{resp}. This is done by evaluating and comparing the performance of an RPL-based network under DIS attacks while applying three modes; DIO_{resp}, DIS_{thr}, and the standard RPL with no mitigation referred to as NON.

A. Simulation Setup

This evaluation considers a grid topology of 50 static nodes for the network under study as shown in Fig. 3. There are many IoT applications that consider grid-based deployment. However, different topologies such as random or linear topology can be considered for experimentation in this study seamlessly. The border router (i.e., sink) is the node with the label 1 and all other nodes, including the attackers, are ordinary nodes. The green and gray areas show the transmission and interference range of the node ID#23, respectively. These ranges are set at 30 m and 40 m and are unified for all nodes in the network. With these configurations, the neighbor density is eight nodes. The Skymote platform is used in this evaluation. The network performance is evaluated under two scenarios; when 10% and 20% of the total nodes act as attackers. The attacker nodes are distributed randomly across the network. For each scenario, the simulation is repeated 10 times and the results are averaged across all runs.

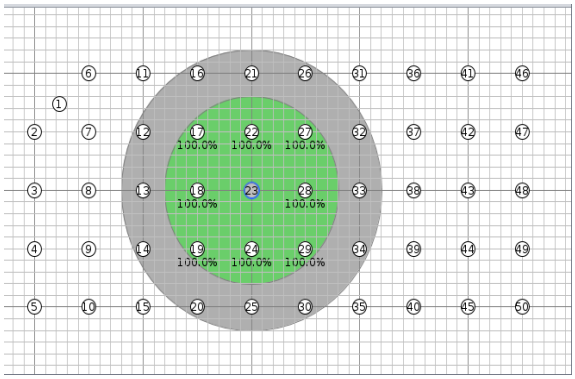

Fig. 3. The network topology under study

Table I summarizes the parameters used throughout this simulation.

TABLE I
Configurations for Experiments

Simulation Tool	Contiki 3.0 and Cooja Simulator
Topology	2D-Grid
Mote Type	Sky Mote
Simulation Run Time	30 Minutes
Network Size	50 Nodes
Mode of operation	Non-storing mode
MAC & PHY protocols	CSMA/CA & IEEE 802.15.4
Radio Medium	Unit Disk Graph Medium (UDGM)
Transmission Range	$80m$
Interference Range	$120m$
Number of Iterations	10
Start Delay (T_d)	5 Seconds
DIS_{thr}	5
DIO_{thr}^{resp}	5

B. Results on Routing Control Traffic (RCT)

RCT is the total number of control messages (i.e. DIOs, DAOs, and DISs) sent by all nodes in the network throughout the simulation. Figure 4 shows that DIO_{resp} mode generates less RCT than NON and DIS_{thr} modes for both attacker percentages (i.e. 10% and 20%). For the 10% scenario, DIO_{resp} decreases RCT by 35.42% and 7.8% relative to the NON and DIS_{thr} modes, respectively. Also, it can be seen that RCT reduction is greater for a greater percentage of attackers (i.e. 20%) with an average reduction of 39% and 9.4% compared to the NON and DIS_{thr} modes, respectively.

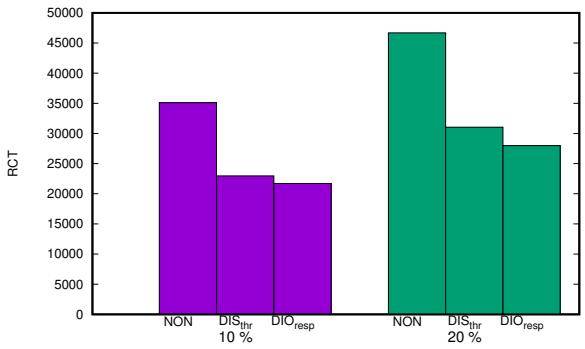

Fig. 4. RCT as a function of attacker percentage

For further analysis, Fig.5 breaks down RCT into its main components: DIOs, DAOs, and DISs when both attacker percentages are applied. One can see that the DIO_{resp} decreases all RPL-enabling messages compared to other modes. For the 10% scenario, DIO and DAO messages are decreased by 12.7% and 9.3%, respectively, compared to DIS_{thr}. However, they are decreased by 16.4% and 15%, respectively, when 20% of nodes are set as attackers. As expected, the number of DIS messages is the same for all modes in a particular scenario. This is because sending DIS messages is a user-defined parameter that is set at the beginning of the experiment. Finally, to serve as a comparison, this figure also illustrates control messages related to NON mode.

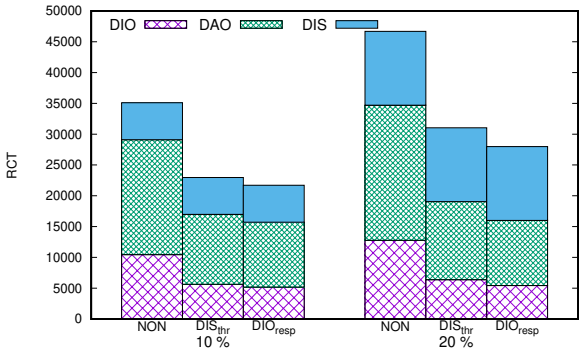

Fig. 5. Breakdown of RCT to its main components

As expected, all nodes in the network would be impacted by the DIS attack in terms of the number of control messages that they should transmit. To show this impact, Figs. 6 and 7 depicts the number of transmitted DIO and DAO messages, respectively, per each node for 20% scenario when NON, DIS_{thr}, and DIO_{resp}, are used. It can be observed that all nodes transmit DIO and DAO messages at a higher rate in the case of the NON mode relative to the DIS_{thr} and DIO_{resp} modes. Generally, the closer the nodes are to attackers, the higher the number of messages emitted as these nodes are susceptible to frequent resetting events. Additionally, figures show that the DIS_{thr} and DIO_{resp} modes manage to reduce DIO and DAO transmissions. Finally, the DIO_{resp} mode gives better results compared to the DIS_{thr} mode.

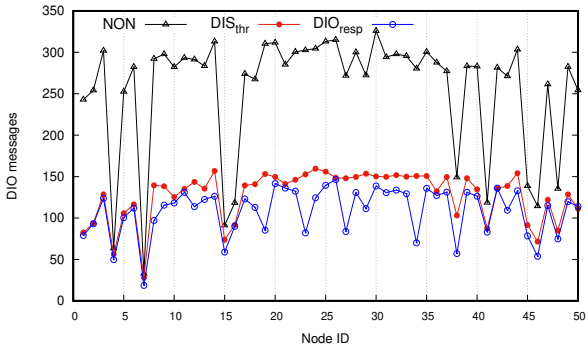

Fig. 6. Number of DIOs per node for 20% attacker percentage

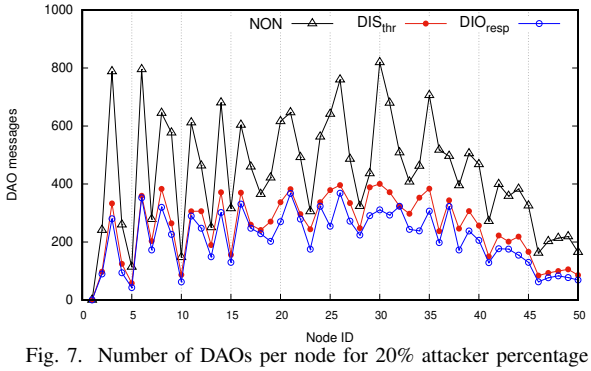

Fig. 7. Number of DAOs per node for 20% attacker percentage

C. Results on Packet Delivery Ration (PDR)

The PDR is the fraction of data packets successfully received over those sent in end-to-end communication. Figure 8 depicts the performance of the considered modes in terms of the PDR percentage under attacker percentages of 10% and 20%. The following observations can be derived from the figure. Firstly, the DIS attack causes detrimental effects on the PDR. This degradation can be mainly justified by the congestion brought by the high overhead at attack-affected nodes which, in turn, has been alleviated by applying mitigation mechanisms. Secondly, both the DIS_{thr} and DIO_{resp} modes effectively enhance the network performance under the DIS attack in terms of the PDR. Finally, DIO_{resp} provides slightly higher PDR percentages that are about 1.2% and 2.01% more compared to that for DIS_{thr} under attacker percentages of 10% and 20%, respectively. Such improvement is greater compared to NON mode. This is because the high number of control messages sent in the network under attack would increase the network congestion and buffer overflow which cause a high placket loss rate [14]

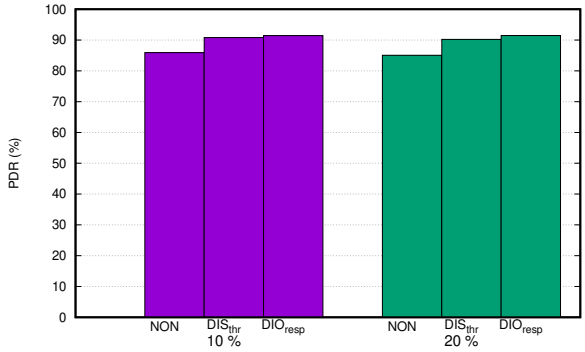

Fig. 8. PDR using studied modes as a function of attacker percentage

D. Results on Energy Consumption

This subsection evaluates the network performance in terms of the consumed energy when NON, DIS_{thr}, and DIO_{resp} modes are utilized. We report the energy consumption of the whole network and for individual nodes as well. Figure 9 depicts the energy consumption for the whole network using the considered modes as a function of both attacker percentages. From this figure, the following observations can be inferred. Firstly, increasing the energy consumption in the NON mode emphasizes the adverse effect of the attack on the network as a whole which is a byproduct of a large number of RCT transmitted in the network due to the attack. Secondly, the energy consumption is effectively decreased in both the DIS_{thr} and DIO_{resp} modes compared to the NON mode. Specifically, this decrease is an average of 20.9% and 22.9%, respectively, relative to the NON mode for attacker percentage of 10%. This decrease is greater for the attacker percentage of 20% as it is clear from the figure. Finally, DIO_{resp} shows a relatively better performance in terms of energy compared to DIS_{thr} under both the 10%

and 20% attacker percentages. DIO_{resp} decreases energy consumption by an average of 2.04% and 7.4% compared to the DIS_{thr} mode under attacker percentages of 10% and 20%, respectively. In summary, these findings demonstrate that

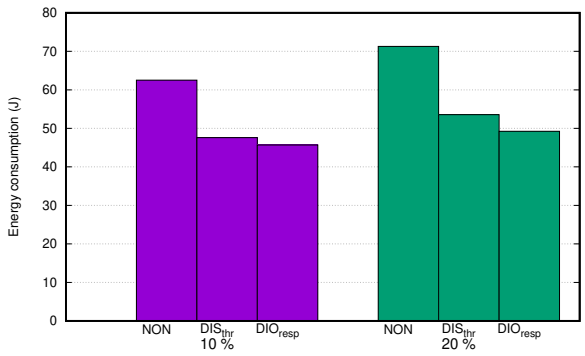

Fig. 9. Total network energy usage as a function of 10% and 20% attacker percentages

DIO_{resp} is feasible and effective when it comes to energy consumption. This is attributed to the DIO_{resp} managing to reduce the overall RCT transmissions in the network.

For further analysis, Fig. 10 depicts the energy consumption per node using the considered modes under an attacker percentage of 20%. Due to the high rate of RCT transmissions experienced by all nodes, energy consumption increased accordingly. This can be clearly observed when the NON mode is used. In general, the closer the node is to the attackers, the greater the energy consumption. However, the figure demonstrates that when DIS_{thr} and DIO_{resp} modes are used, energy consumption decreases for each node individually. Moreover, when compared to the other two modes, the DIO_{resp} mode offers further energy consumption improvements across all nodes.

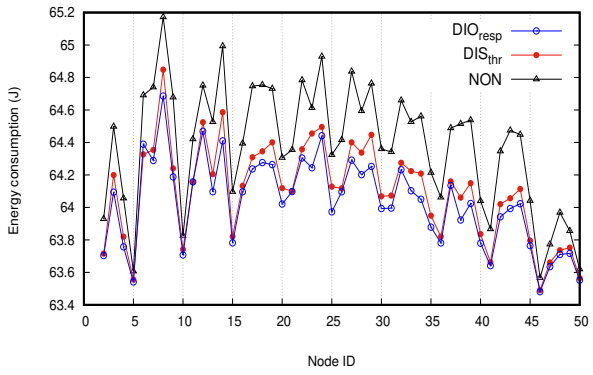

Fig. 10. Energy usage per node as a function of 20% attacker percentage

VI. Conclusion

In this study, the negative effect of DIS attacks on RPL-based networks is addressed. The DIS attack can be mounted in these networks by having an attacker node transmit DIS messages periodically to its neighbors, which in turn triggers receiving nodes to respond by sending DIO messages. From the simulation results obtained, we demonstrated that DIS attacks lead to an increase in RCT and energy consumption, which significantly degrades the network performance. To overcome the attack's effect, we proposed and evaluated a mitigation technique referred to as DIO_{resp}. The proposed technique manages to guard against DIS attacks and shows a good capacity for improving the routing performance of the studied network in terms of RCT, PDR, and energy consumption.

References

[1] ITU, "Overview of the internet of things," *Recommendation ITU-T Y.2060*, 2012.

[2] F. Medjek, D. Tandjaoui, N. Djedjig, and I. Romdhani, "Multicast dis attack mitigation in rpl-based iot-llns," *Journal of Information Security and Applications*, vol. 61, p. 102939, 2021.

[3] I. Lee and K. Lee, "The internet of things (iot): Applications, investments, and challenges for enterprises," *Business Horizons*, vol. 58, no. 4, pp. 431–440, 2015.

[4] T. Winter, P. Thubert, A. Brandt, T. H. Clausen, J. W. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, and J. Vasseur, "RPL: IPv6 Routing Protocol for Low Power and Lossy Networks," *Network Architectures and Services*, no. July, pp. 59–66, 2011.

[5] U. Farooq, M. Asim, N. Tariq, T. Baker, and A. I. Awad, "Multi-mobile agent trust framework for mitigating internal attacks and augmenting rpl security," *Sensors*, vol. 22, no. 12, 2022.

[6] Nisha, A. Dhingra, and V. Sindhu, "A review of dis-flooding attacks in rpl based iot network," in *2022 International Conference on Communication, Computing and Internet of Things (IC3IoT)*, 2022, pp. 1–6.

[7] T. Winter, P. Thubert, A. R. Corporation, and R. Kelsey, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks," *RFC 6550*, pp. 1–157, 2012.

[8] X. Niu, "Optimizing dodag build with rpl protocol," *Mathematical Problems in Engineering*, vol. 2021, 2021.

[9] P. Levis, N. Patel, D. Culler, S. Shenker, and D. Culler, "Trickle: A self-regulating algorithm for code propagation and maintenance in wireless sensor networks," *Proceedings of the First Symposium on Networked Systems Design and Implementation*, pp. 15–28, 2004.

[10] M. Mahyoub, A. S. Hasan Mahmoud, M. Abu-Amara, and T. R. Sheltami, "An efficient rpl-based mechanism for node-to-node communications in iot," *IEEE Internet of Things Journal*, vol. 8, no. 9, pp. 7152–7169, 2021.

[11] A. Verma and V. Ranga, "Mitigation of DIS flooding attacks in RPL-based 6LoWPAN networks," *Transactions on Emerging Telecommunications Technologies*, vol. 31, no. 2, pp. 1–25, 2020.

[12] B. Farzaneh, M. A. Montazeri, and S. Jamali, "An anomaly-based ids for detecting attacks in rpl-based internet of things," in *2019 5th International Conference on Web Research (ICWR)*, 2019, pp. 61–66.

[13] G. Guo, "A lightweight countermeasure to dis attack in rpl routing protocol," in *2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC)*, 2021, pp. 0753–0758.

[14] H. A. A. Al-Kashoash, H. M. Amer, L. Mihaylova, and A. H. Kemp, "Optimization-based hybrid congestion alleviation for 6lowpan networks," *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 2070–2081, 2017.