Addressing Flooding Attacks in IPv6-based Low Power and Lossy Networks

Abhishek Verma [1] and Virender Ranga [2]

[1]National Institute of Technology Kurukshetra
[2]Affiliation not available

October 30, 2023

Abstract

Abstarct: In the RPL routing protocol, DODAG Information

Solicitation (DIS) control messages are sent by nodes to join the network. In turn, the receiver node replies with DODAG Information Object (DIO) control message after resetting its trickle timer. A malicious node can utilize this RPL protocol behavior to perform the DIS flooding attack by sending illegitimate DIS frequently which forces normal nodes to reset their trickle timers and flood the network with DIO messages. In this study, we show that such attacks can severely degrade the performance of Low Power and Lossy Networks (LLNs) because of the increase in control packet overhead and power consumption. To address DIS flooding attacks, we propose a lightweight mitigation scheme that detects and mitigate such attacks in order to improve LLNs

performance.

Note:To be published in proceedings of 2019 IEEE Region 10 Conference (TENCON 2019)

Addressing Flooding Attacks in IPv6-based Low Power and Lossy Networks

Abhishek Verma
Department of Computer Engineering
National Institute of Technology
Kurukshetra, India
abhiverma866@gmail.com

Virender Ranga
Department of Computer Engineering
National Institute of Technology
Kurukshetra, India
virender.ranga@nitkkr.ac.in

***Abstract*—In the RPL routing protocol, DODAG Information Solicitation (DIS) control messages are sent by nodes to join the network. In turn, the receiver node replies with DODAG Information Object (DIO) control message after resetting its trickle timer. A malicious node can utilize this RPL protocol behavior to perform the DIS flooding attack by sending illegitimate DIS frequently which forces normal nodes to reset their trickle timers and flood the network with DIO messages. In this study, we show that such attacks can severely degrade the performance of Low Power and Lossy Networks (LLNs) because of the increase in control packet overhead and power consumption. To address DIS flooding attacks, we propose a lightweight mitigation scheme that detects and mitigate such attacks in order to improve LLNs performance.**

***Index Terms*—Internet of Things, DIS flooding, RPL, 6LoWPAN, LLN**

I. Introduction

The applications of the Internet of Things (IoT) [1] are now considered as key players in making human life easier. The critical IoT applications like smart healthcare, smart home and industrial monitoring have proven to be very effective for personal purposes and commercial organizations. However, with many benefits, IoT also has major security and privacy risks. The vulnerabilities of supporting devices, communication technology, and smart applications can be exploited by the attacker in order to either perform Denial of Services attack or gain access to user's private information [2]. In present, most of the IoT applications require an infrastructure that supports longer operation time and has minimum implementation cost. Low Power and Lossy Networks (LLNs) [3] provide such infrastructure with desired operation time and cost. LLNs consist of resource constrained (memory, processing, communication and energy) devices (nodes) which operate on standard IoT protocol stack. Resource constrained nodes need an energy efficient network layer protocol for routing packets in the network. IETF's RoLL group conceptualized a routing protocol named Routing Protocol for Low-power Lossy Networks (RPL) [4] for such purposes. RPL has now become a standard routing protocol for IoT applications and its specifications are presented in RFC 6550. The characteristics like self-organization, self-healing, and open nature expose RPL to insider and outsider threats [5]. Meanwhile, the RPL protocol incorporates a secure mode to provide defense against various attacks. The secure mode uses traditional cryptography to maintain integrity and confidentiality of the data. However, the current standard does not specify the key management methods for ensuring proper distribution of security keys among nodes [6]. The traditional defense mechanisms that use cryptography techniques are unsuitable for resource constrained devices because of high computational and memory requirements [7].

In the typical implementation of RPL protocol, the secure mode is not recommended because of energy constraints. The insecure mode of RPL is vulnerable to different attacks where a malicious node may exploit RPL supporting mechanisms to target smart devices, consequently bringing down overall network performance. DODAG Information Solicitation (DIS) flooding attack is one of such attacks in which an attacker targets IoT network resources directly. In DIS flooding attack, a malicious outsider or insider node periodically sends DIS control messages to the nodes within its range. This forces receiver nodes to flood the network with DODAG Information Object (DIO) messages. This situation can drastically affect the critical network parameters like control packet overhead, power consumption, latency and reliability. RPL does not have any inbuilt defense mechanism to provide security against such attacks, thus the resource constrained nature of 6LoWPANs may even become more vulnerable to flooding attacks [8]. In case of fast flood rate, the non-attacker nodes are forced to reset their trickle timer and send DIOs frequently which drastically increases control packet overhead and node power consumption. Whereas in case of slow flood rate, the network performance is less impacted. Our contributions include a study on impact analysis of DIS flooding attacks on RPL based LLNs. Based on the study, we propose a mitigation scheme for detecting and mitigating DIS flooding attacks.

The paper is organized in the following manner. In Section II, related works are discussed. , followed by a brief introduction of the RPL protocol in Section III. The DIS flooding attack is over-viewed in Section IV, followed by a discussion on the proposed mitigation scheme in Section V. An experimental evaluation of the proposed mitigation scheme is done in Section VI. Conclusion and future work are discussed in Section VII.

978-1-7281-1895-6/19/$31.00 ©2019 IEEE

II. Related Work

The studies related to detection and mitigation of flooding attacks have been done in particular to Wireless Sensor Networks. However, Wireless Sensor Networks (WSN) specific security solutions cannot be directly applied to RPL based networks because of different protocol design and control messages involved. There are several works present in the literature that focus on DIS flooding attacks. Le *et al.* [9] investigated the impact of rank, local repair, replay and DIS flooding attacks on RPL protocol. The authors showed that the DIS flooding attack has the highest impact on network performance. Le *et al.* [10] proposed an Intrusion Detection System (IDS) to detect routing attacks. Due to additional overhead and inefficient longer operation, the available solution is unsuitable for RPL networks. Different routing attacks specific to RPL have also been addressed in the literature. Like, Ghaleb *et al.* [11] studied the impact of Destination Advertisement Object (DAO) falsification attack and proposed a defense mechanism named SecRPL to mitigate the same. Ariehrour *et al.* [12] proposed an enhancement to RPL which is capable of detecting rank, and sybil attack. A hybrid of anomaly and specification based IDS to detect sinkhole and selective forwarding attacks is proposed by Bostani *et al.* [13]. Verma *et al.* [14] developed a dataset for evaluation of anomaly based network IDS in particular to RPL protocol. The authors simulated different routing attacks including flooding attack for collecting network traffic. They also tested the performance of different machine learning classifiers over the developed dataset.

III. RPL Protocol

The RPL protocol [4] is based on the distance-vector and source routing protocols and operates on top of standard IEEE 802.15.4 protocol. It supports point-to-point, multipoint-to-point and point-to-multipoint topology. RPL forms a Destination Oriented Directed Acyclic Graph (DODAG) from IoT nodes. A DODAG consists of nodes (i.e., router, host and gateway) which organize themselves into a particular form of topological structure in order to carry out routing in LLNs. A single IoT network contains multiple parallel RPLInstance running at a single time, and a single RPL Instance may contain multiple DODAGs. RPL Instance is identified by RPLInstanceID while DODAG is identified by DODAG ID which is a unique IPv6 address. The main characteristics of the RPL protocol include auto-configuration, self-healing, loop avoidance and detection, transparency, and support for multiple edge routers or sink. RPL uses four types of control messages (DIO, DIS, DAO, DAO-ACK) for creating and maintaining DODAG. Routes between DODAG nodes are selected and optimized using an Objective function (OF). An OF uses various metrics and constraints in order to select the optimal path and parent among different preferred choices. Nodes are assigned a rank value (16 bit) which represents the nodes individual position with respect to DODAG root. The rank concept is used to maintain the parent-child relationship, as well as to prevent loops in the network.

IV. DIS Flooding Attacks

In RPL, DIS messages are used by nodes to join the network. A node sends a DIS message to its neighbor nodes in order to request the routing information so that it may join the existing DODAG. RFC 6650 does not specify the time interval for DIS transmission. The only information provided in this regard is that the DISs are used to request the DIO that contains the information regarding existing DODAG. Thus, the DIS transmission interval may vary with different implementations. The most poplar RPL implementation, i.e., Contiki considers 60 seconds as the fixed DIS transmission interval. Thus, a new node continuously transmits DISs with a fixed interval of 60 seconds until it receives a DIO message from any neighbor node. Once a node receives a DIO message, it stops transmitting DIS messages and joins the network by sending DAO to the solicited-node. In Contiki RPL, a node can either multicast or it can unicast DIS message. Based on the nature of transmission (i.e., multicast or unicast), the receiver responds in the following manner:

- Case 1: On receiving a multicast DIS, the receiver node resets trickle timer and sends multicast DIO messages containing the latest routing information.
- Case 2: On receiving a unicast DIS, the receiver node directly sends DIO message to the sender node without resetting trickle timer.

A malicious node can utilize this feature to degrade the network performance by choosing different DIS transmission interval for periodically transmitting DIS messages to its neighboring nodes; this is called a DIS flooding attack. In such a situation, non-attacker receiver node is forced to respond as per the mentioned case 1 or 2. This leads to an increase in the network's control packet overhead and power consumption.

V. Proposed Mitigation Scheme

In this section, we present the proposed mitigation scheme. In order to design the anti-DIS flooding mechanism, we studied the normal DIS transmission mechanism of RPL. We observe that a node sends DIS message in two situations: 1) when a node wants to join the network it sends a DIS to request active DODAG information; 2) when a node loses all links from its current DODAG. According to RPL specification [4], a node is set with two DIS specific parameters (constants) which govern the transmission of DIS messages. First is *DIS_START_DELAY*, it represents the time a node must wait before sending first DIS. The second parameter is *RPL_DIS_INTERVAL* which represents the time interval between two consecutive DIS messages. In most popular RPL implementation, i.e., ContikiRPL, the values of *DIS_START_DELAY* , *RPL_DIS_INTERVAL* are 5 second and 60 seconds respectively. RPL specifies that a disconnected node can keep on transmitting DIS messages in set *RPL_DIS_INTERVAL* until it receives a DIO message. Moreover, upon the reception of DIS, the receiver node has to reply with DIO message. This feature of RPL can be exploited by an attacker that may target the nodes from inside or outside

the network. Based on the values of *DIS_START_DELAY*, *RPL_DIS_INTERVAL* we choose two safety thresholds α and β that control the trickle timer resets of the nodes. We set the value of α, β to 60 and 5 respectively. α corresponds to safe DIS transmission interval, β corresponds to the maximum permitted DIS requests. α provides defense against fast DIS flooding attacks while β provides defense against slow DIS flooding attacks. To be more specific, we used predefined RPL parameters for designing the proposed mitigation scheme. Pseudo-code of the proposed mitigation scheme is shown in Algorithm 1. The mitigation scheme is executed every time a node receives a DIS message. It consists of four core procedures namely INITIALIZATION, NODE-ALLOCATION, SEARCH-BLACKLIST, and DIS-RECEIVED. Upon initialization of RPL modules, the INITIALIZATION procedure (lines $1-3$) initializes the defense mechanism specific variables and structures. The variable max_nodes maintains the count of the active nodes in the network, $t_{current}$ stores the current system time. Moreover, two arrays namely $blacklist$, $node\_table$ are maintained to store blacklisted nodes, and node information respectively. The node information (entry in $node\_table$) is maintained using a user-defined structure of three elements, i.e., $[<from, timestamp, count_{DIS}>]$. Where, $from$ stores DIS sender,$timestamp$ stores time of DIS receipt, and $count_{DIS}$ is the counter to maintain the count of DIS received from the particular node. The NODE-ALLOCATION procedure (lines $4-10$) initializes the $node\_table$ when the mitigation is executed for the first time since node start. The mitigation scheme maintains a list of blacklisted nodes which is searched upon receiving a DIS message. The $blacklist$ is searched in SEARCH-BLACKLIST procedure (lines $11-18$). The main procedure is DIS-RECEIVED which is Incorporated in the default ContikiRPL's "dis_input" method. DIS-RECEIVED starts by storing the time of DIS receipt in $t_{current}$ (line 20). Then, NODE-ALLOCATION procedure is executed if the procedure is called for the first time since node startup (lines $21-23$). The DIS sender is searched in $blacklist$ (line 24). Then the sender node's entry is searched in $node\_table$. In case the node's entry is found, the violation of safety threshold limits is checked. If the DIS sender violates the safety limits, the DIS is discarded and the sender is added to $blacklist$ (lines $25-30$). Otherwise, the sender information is updated in $node\_table$ (lines $33-34$). If the sender node is not present in the $node\_table$ (lines $25-26$), then its latest information is added to the $node\_table$ (lines $37-45$).

Algorithm 1 Pseudo-code of proposed mitigation scheme

```
1: procedure INITIALIZATION
2:     max_nodes, blacklist, node_table, t_current, α, β,
   node ← [< from, timestamp, count_DIS >]
3: end procedure
4: procedure NODE-ALLOCATION
5:     for i ← 0 to max_nodes do
6:         node_table[node_i.from] ← null_IP
7:         node_table[node_i.timestamp] ← 0
8:         node_table[node_i.count_DIS] ← 0
9:     end for
10: end procedure
11: procedure SEARCH-BLACKLIST
12:     for each node in blacklist do
13:         if blacklist[node] = sender_IP then
14:             discard the DIS
15:             return
16:         end if
17:     end for
18: end procedure
19: procedure DIS-RECEIVED
20:     get t_current from system clock
21:     if node_table is empty then
22:         execute NODE-ALLOCATION
23:     end if
24:     execute SEARCH-BLACKLIST(sender_IP)
25:     for each node in node_table do
26:         if node_table[node.from]=sender_IP then
27:             if t_current − node_table[node.timestamp] < α
   OR node_table[node.count_DIS] > β then
28:                 discard the DIS
29:                 store sender_IP to blacklist
30:                 return
31:             end if
32:             InTable_flag ← 1
33:             set node_table[node.timestamp] to t_current
34:             Increment node_table[node.count_DIS]
35:         end if
36:     end for
37:     if InTable_flag = 0 then
38:         for i ← 0 to max_nodes do
39:             if node_table[node_i.from]=null_IP then
40:                 set node_table[node_i.from] to sender_IP
41:                 set node_table[node_i.timestamp] to t_current
42:                 Increment node_table[node_i.count_DIS]
43:             end if
44:         end for
45:     end if
46: end procedure
```

VI. Results and Discussion

To evaluate the performance of proposed mitigation scheme we used Cooja simulator of Contiki 3.0 running on Linux 18.14 (64 bit) and operated on a machine equipped with Intel® i7-7700 four core CPU having 3.60 GHz clock speed and 12 GB main memory. Contiki incorporates well tested standard fundamental mechanisms of RPL protocol which are compatible with different hardware platforms. Cooja runs a hardware simulator named MSPSim that emulates exact binaries of real sensor nodes. In this paper, we used Zolertia 1 (Z1) platform (MSP430 architecture based ultra-low power microcontroller board) which has the IEEE 802.15.4 compliant CC2420 radio transceiver operating at 2.4 GHz. The Power-

tracker tool of Cooja is used to log radio transceiver power statistics of each node in terms of radio on (ON), transmitting (TX), receiving (RX), and interfered (INT). The msp430-size tool is used to study the memory requirements of proposed mitigation scheme. The performance evaluation is done of control packet overhead (i.e., total number of control packets required to maintain the topology), and power consumption (radio event times). A network scenario containing 16 nodes as shown in Fig. 1 is simulated. It consists of one 6LoWPAN border router (6BR), one malicious node, and 14 non-attacker sensor nodes. The simulation parameters are listed in Table I. We have positioned the malicious node in close proximity of 6BR in order to make maximum impact on the network. The simulation is repeated thirty times for accuracy reasons. We compare the performances of standard MRHOF-RPL (i.e., RPL with Minimum Rank with Hysteresis Objective Function and no attack scenario), Insecure-RPL (i.e., RPL under attack), and Secure-RPL (i.e., RPL under attack with our proposed mitigation scheme). It is to be noted that the MRHOF-RPL is not any existing approach, its just a normal (not under attack) RPL protocol with MRHOF as a objective function. Our main goal is to minimize the affect of flooding attacks on Insecure-RPL protocol such that the difference between normal (MRHOF-RPL) and Secure-RPL is minimum.

TABLE I
SIMULATION PARAMETERS

Parameter	Values
Routing protocol	RPL
MAC protocol	IEEE 802.15.4
Radio model	UDGM
Simulation grid	200 × 200 meters
Simulation time	5, 10, 15 minutes
Objective function	MRHOF
Attacker nodes	1
Transmission (TX) range	50 meters
Interference (INT) range	100 meters
Sensor nodes	16
DIO minimum interval (I_{min})	4 seconds
DIO maximum interval (I_{max})	17.5 minutes

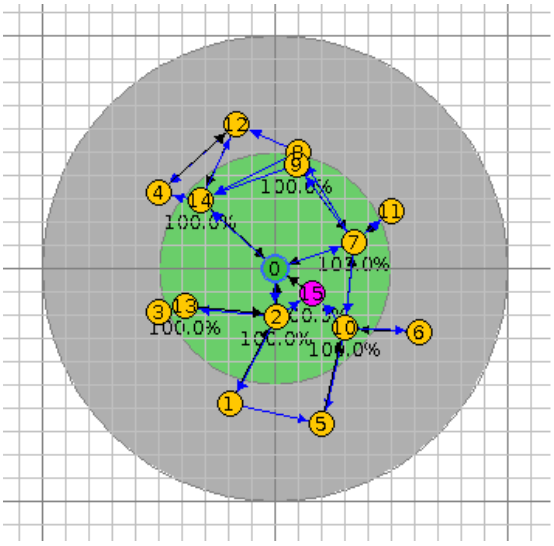

Fig. 1. Network scenario (●, ●, ● represent 6BR, non-attacker, and malicious node respectively.)

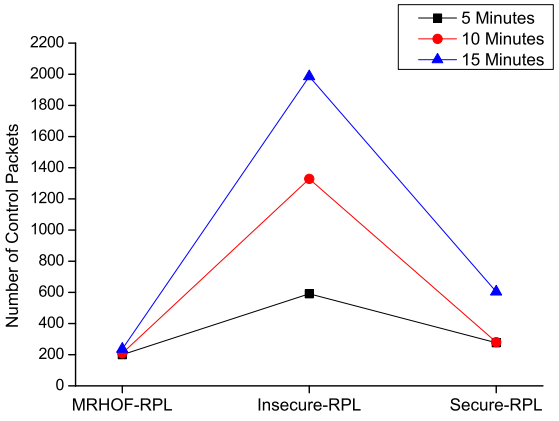

Fig. 2. Control packet overhead

A. Control packet overhead

The control packet overhead is calculated for three different simulation times (i.e., 5, 10, 15 minutes) and the results are presented in Fig. 2. The impact of DIS flooding attack on the network can be seen from the experimental results. In the case of Insecure-RPL, the control packet overhead of the network is very high as compared to MRHOF-RPL because of the increase in the number of DAO and DIO transmissions. The standard MRHOF-RPL registers the minimum control packet overhead for all three runs whereas the Insecure-RPL shows the worst results for each simulation. Our proposed mitigation scheme is able to minimize the control packet overhead of the network by minimizing trickle timer resets that consequently decreases unnecessary DIO transmissions.

In network scenario shown in Fig. 1, it can be observed that the nodes 0, 2, 7, and 10 (Node ID) are in close proximity of malicious node 15. The nodes 6 and 11 also come in the attacker's transmission range but not in close proximity. The nodes in close proximity of attacker are more likely to be impacted by flooding attack while distant nodes may have less impact. The same is observed in Fig. 3, where various control messages sent by non-attacker nodes are shown. In the case of Insecure-RPL, nodes 0, 2, 7, and 10 register increase in the number of DIO and DAO messages as compared to MRHOF-RPL. Moreover, other nodes also show an increase in DAO transmissions. The reason is that the congestion caused by DIO and DIS messages leads to a collision of DAO messages which are then re-transmitted. In Secure-RPL case, our proposed mitigation scheme reduces the impact of DIS flooding attack by controlling the unnecessary trickle timer resets. The results of 5 and 10 minutes simulation are not included in the paper because of page constraints.

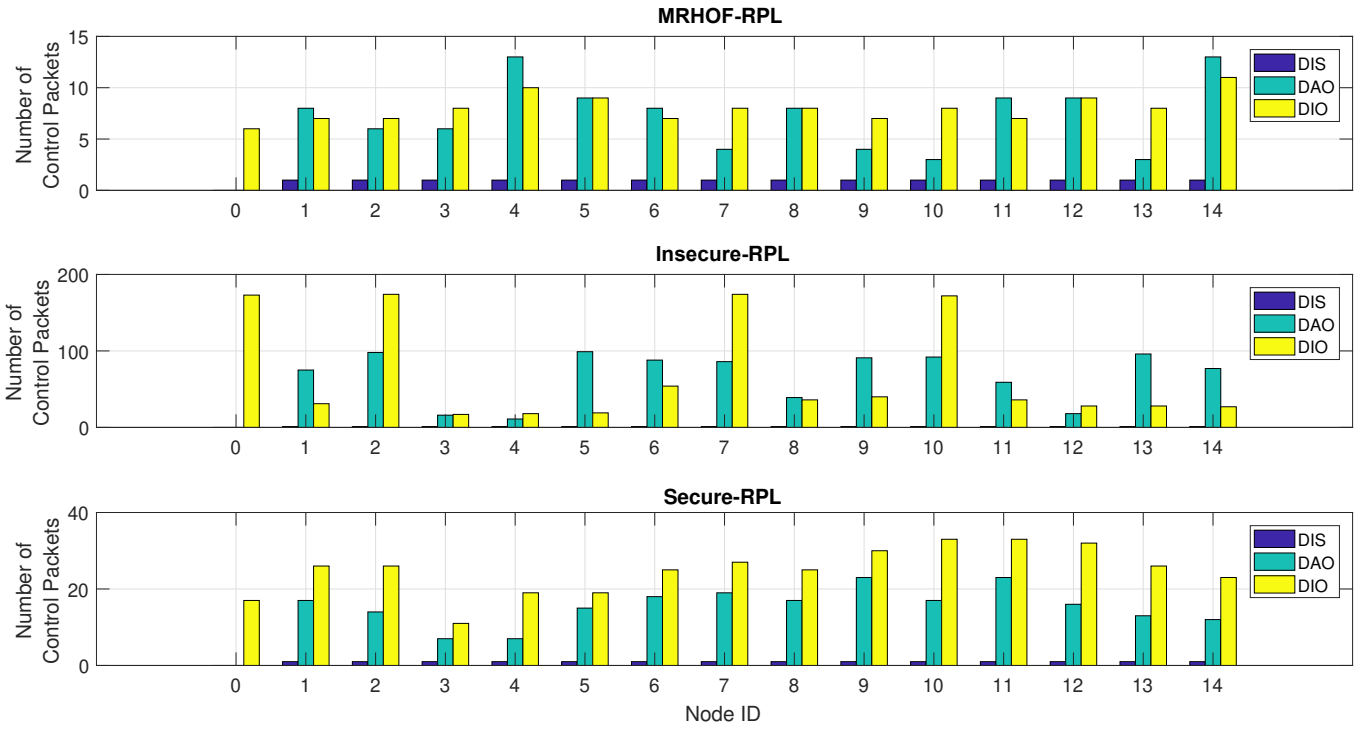

Fig. 3. Number of DIS, DIO, and DAO messages transmitted (15 minutes)

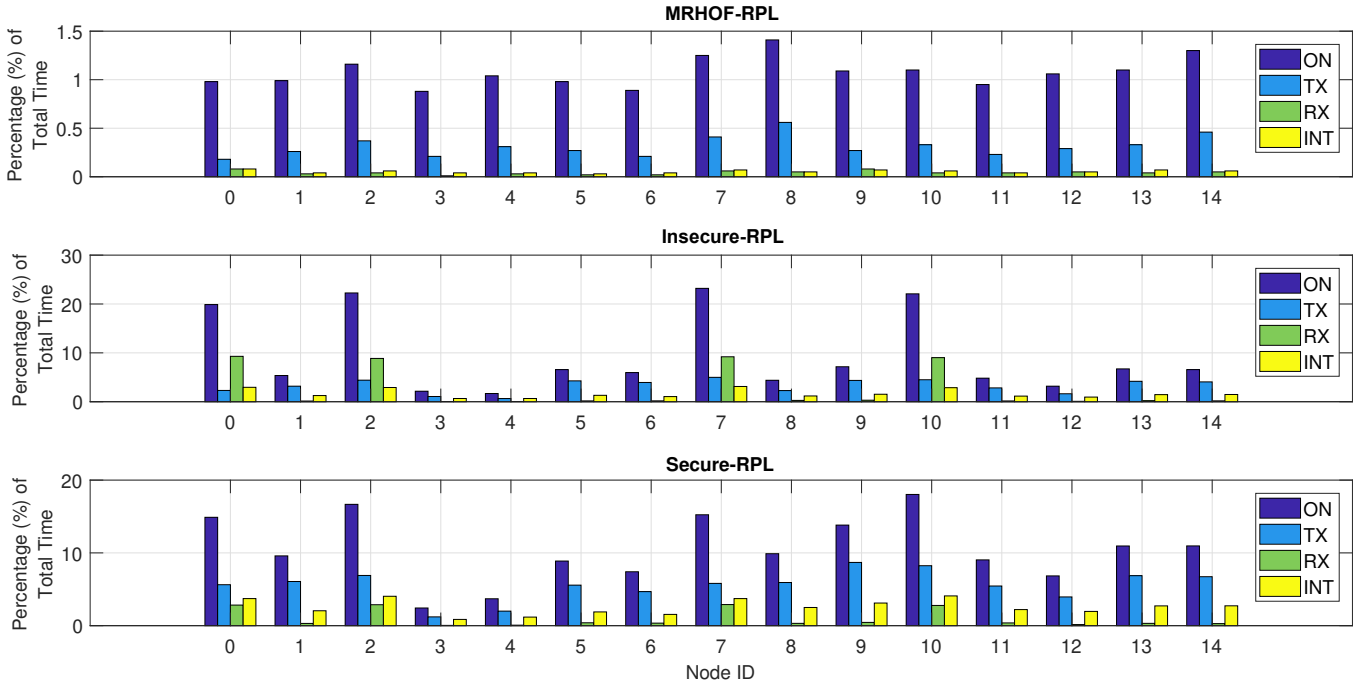

Fig. 4. Power consumption in terms of radio event times (15 minutes)

B. Power consumption

In order to study the power consumption of nodes, we analyze the total time spent by nodes in different radio states like ON, TX, RX, and INT. Fig. 4 shows different radio event times for each node. The node radio spends most of the time in ON and TX state under no attack condition as shown in the case of MRHOF-RPL. The nodes $0, 2, 7$, and 10 registered highest ON and RX times as compared to maximum ON and RX times of MRHOF-RPL. This difference represents the impact of the attack on the nodes. The proposed mitigation scheme efficiently mitigates the effect of the attack by significantly bringing down ON and RX times of impacted nodes. In Secure-RPL, ON and RX time is reduced because of the mitigation scheme which discards malicious DIS and prevents nodes from unnecessary trickle timer resets. The experimental results of power consumption shown in Fig. 4 are in sync with results of control packet overhead evaluation shown in Fig. 2.

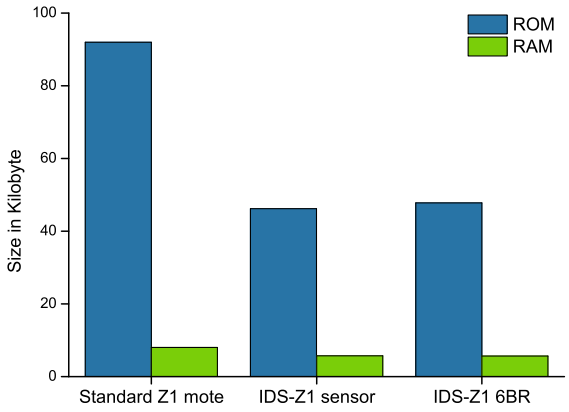

Fig. 5. Memory consumption

C. Implementation overhead

Resource constraints of LLNs restrict the usage of resource-hungry security mechanisms like cryptography. Hence, it is very important to design defense solution that fits in resource constrained nodes and does not add any significant overhead to the network. Fig. 5 depicts memory footprints of the proposed mitigation scheme on sensor and 6BR. The standard Z1 mote (i.e., node) is equipped with 8 kB RAM and 92 kB ROM. Whereas, the Z1 binary (executable binary containing base Contiki system files that run on Z1 mote) with proposed mitigation scheme that runs on sensor node needs 46.16 kB of ROM and 5.75 kB of RAM. Similarly, the Z1 binary that runs on 6BR requires 47.77 kB of ROM and 5.66 kB of RAM. Our proposed mitigation scheme only needs an additional 0.498 kB of ROM and 0.654 kB of RAM. Thus, the proposed mitigation scheme can easily run on resource constrained nodes of LLNs with little overhead.

VII. Conclusions and Future work

Most of the IoT applications are based on LLNs due to longer operation requirement. LLNs are vulnerable to various insider and outsider attacks, thus ensuring security and privacy of such networks is the most important task. In this paper, we studied the effect of DIS flooding attack on network performance in terms of prominent evaluation metrics. We presented a mitigation scheme to defend LLNs against DIS flooding attacks. The experimental results show that our proposed mitigation scheme mitigates such attack while significantly improving network performance without significant implementation overhead. Our future plan is to study the DIO suppression attacks and design a mitigation scheme to defend such attacks.

Acknowledgments

This research was financially supported by the Ministry of Human Resource Development (MHRD), Government of India and TEQIP III.

References

[1] L. Da Xu, W. He, and S. Li, "Internet of things in industries: A survey," *IEEE Transactions on industrial informatics*, vol. 10, no. 4, pp. 2233–2243, 2014.

[2] A. Raoof, A. Matrawy, and C.-H. Lung, "Routing Attacks and Mitigation Methods for RPL-Based Internet of Things," *IEEE Communications Surveys & Tutorials*, 2018, (in press).

[3] J. W. Hui and D. E. Culler, "Extending IP to low-power, wireless personal area networks," *IEEE Internet Computing*, no. 4, pp. 37–45, 2008.

[4] T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, J.-P. Vasseur, and R. Alexander, "RPL: IPv6 routing protocol for low-power and lossy networks," Tech. Rep., 2012.

[5] L. Wallgren, S. Raza, and T. Voigt, "Routing Attacks and Countermeasures in the RPL-based Internet of Things," *International Journal of Distributed Sensor Networks*, vol. 9, no. 8, p. 794326, 2013.

[6] P. Perazzo, C. Vallati, A. Arena, G. Anastasi, and G. Dini, "An implementation and evaluation of the security features of RPL," in *International Conference on Ad-Hoc Networks and Wireless*. Springer, 2017, pp. 63–76.

[7] X. Du and H. H. Chen, "Security in wireless sensor networks," *IEEE Wireless Communications*, vol. 15, no. 4, pp. 60–66, Aug 2008.

[8] T. Nguyen, N. Tri, T. Nguyen, T. Duc, H. A. Tran, and B. Tung, "The Flooding Attack in Low Power and Lossy Networks: A Case Study," in *2018 International Conference on Smart Communications in Network Technologies (SaCoNeT)*. IEEE, 2018, pp. 183–187.

[9] A. Le, J. Loo, Y. Luo, and A. Lasebae, "The impacts of internal threats towards Routing Protocol for Low power and lossy network performance," in *IEEE Symposium on Computers and Communications (ISCC)*, July 2013, pp. 789–794.

[10] A. Le, J. Loo, K. Chai, and M. Aiash, "A specification-based ids for detecting attacks on rpl-based network topology," *Information*, vol. 7, no. 2, p. 25, 2016.

[11] B. Ghaleb, A. Al-Dubai, E. Ekonomou, M. Qasem, I. Romdhani, and L. Mackenzie, "Addressing the DAO Insider Attack in RPLs Internet of Things Networks," *IEEE Communications Letters*, vol. 23, no. 1, pp. 68–71, Jan 2019.

[12] D. Airehrour, J. A. Gutierrez, and S. K. Ray, "SecTrust-RPL: A secure trust-aware RPL routing protocol for Internet of Things," *Future Generation Computer Systems*, vol. 93, pp. 860–876, 2019.

[13] H. Bostani and M. Sheikhan, "Hybrid of anomaly-based and specification-based ids for internet of things using unsupervised opf based on mapreduce approach," *Computer Communications*, vol. 98, pp. 52–71, 2017.

[14] A. Verma and V. Ranga, "Evaluation of Network Intrusion Detection Systems for RPL Based 6LoWPAN Networks in IoT," *Wireless Personal Communications*, pp. 1–24, 2019.