Multicast DIS Attack Mitigation in RPL-Based IoT-LLNs

Faiza Medjek[a,b,*], Djamel Tandjaoui[a], Nabil Djedjig[a,b], Imed Romdhani[c]

[a]*Research Centre on Scientific and Technical Information, 03, Rue des Freres Aissou, Ben Aknoun, Algiers, Algeria*
[b]*Departement Informatique, Faculte des Sciences Exactes, Universite de Bejaia, 06000 Bejaia, Algeria*
[c]*Edinburgh Napier University, School of Computing, 10 Colinton Road, EH10 5DT, Edinburgh, UK*

Abstract

The IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL) was standardised by the IETF ROLL Working Group to address the routing issue in the Internet of Things (IoT) Low-Power and Lossy Networks (LLNs). However, RPL-based LLNs are vulnerable to various attacks because of the resource-constrained nature of LLNs, the lack of tamper resistance, and the security features of RPL. DODAG Information Solicitation (DIS) are ICMPv6 control messages sent by a node intending to join the Destination-Oriented Directed Acyclic Graph (DODAG). Malicious nodes can exploit this mechanism to trigger an attack, named DIS attack against RPL. The DIS attack can have severe consequences on RPL networks, especially on control packets overhead and power consumption. In this paper, we use the Cooja-Contiki simulator to assess the DIS attack's effects on both static and dynamic PRL networks. A novel approach to mitigate RPL against DIS Multicast, namely RPL-MRC, is proposed and implemented. RPL-MRC aims to reduce the response to Multicast messages. Simulation results demonstrated how the attack could damage the network performance by significantly increasing the control packets overhead and power consumption. On the other hand, the proposed mechanism showed a significant enhancement in reducing the control overhead and the power consumption for different scenarios.

Keywords: RPL, RPL Security, Internet of Things, DIS Attack, Routing Attacks, Low Power and Lossy Networks.

1. Introduction

In the Internet of Things (IoT) concept, all physical objects are identifiable, addressable, and interconnected with each other based on standard communication protocol operating the worldwide network [1][2][3]. One of the main building blocks of the IoT is the Low-power and Lossy Networks (LLNs). LLNs are made of a collection of interconnected embedded resource-constrained devices, such as RFID and sensor nodes with low computational and storage capabilities and are often battery operated. In addition, communication technologies are subject to high packet loss, frame size limitations, low data rates, short communication ranges, and dynamically changing network topologies. Such limitations render the development of efficient routing solutions for LLNs crucial [4] [5] [6]. Several attempts have been proposed to handle these issues, like CTP [7], and Hydro [8]. Ultimately, the ROLL IETF Working Group has designed and standardised the Routing Protocol for LLNs, namely the Routing Protocol for Low-Power and Lossy Networks (RPL) [9] [10]. These last years, several studies reported that RPL suffers from security limitations that harm its performances [11] [12].

1.1. RPL Overview

RPL [10] is a distance vector routing protocol that organises the physical network into a logical representation as a Directed Acyclic Graph (DAG) to route data packets. The DAG comprises one or multiple DODAGs (Destination Oriented DAGs) with one root per DODAG. Each root, called a border router (BR), is connected to the Internet, and other potential roots (BRs) via a backbone. Each node in the DODAG has many attributes such as an IPv6 address (ID), a list of parents with one preferred-parent, a list of discovered neighbours and a Rank. The Rank of a node identifies the node's position relative to the BR, respecting the rule that the parent has a lower Rank than the node itself. Specifically, the Rank values should increase from the BR towards the leaf nodes and decrease from the

*Corresponding author
Email address: fmedjek@cerist.dz (Faiza Medjek)

leaf nodes towards the BR. RPL introduces the following ICMPv6 control messages to construct and maintain the network topology and routing paths.

- The DODAG Information Object (DIO) messages convey the relevant information and configuration parameters that enable a node to join a DODAG, select a set of candidate parents, construct and maintain the DODAG. Hence, DIO messages convey node and link metrics and constraints (e.g., node energy, hop count, throughput, latency, link colour, and ETX; Expected Transmission Count) [13], and the Objective Function (OF) [14] [15] to use to optimise the path construction and to calculate the node Rank.
- The DODAG Destination Advertisement Object (DAO) messages allow nodes to propagate their destination information upward along the DODAG to the BR. Consequently, the downward routes from the BR to its associated nodes can be constructed and updated. RPL-DAO messages are transmitted using the end-to-end approach from the nodes to the BR.
- The Destination Advertisement Object Acknowledgement (DAO-ACK) may be unicast by a node to the DAO sender to acknowledge that DAO's reception.
- The DODAG Information Solicitation (DIS) messages aim to discover the neighbourhood and network topology. Precisely, nodes seeking to join a DODAG use DIS messages to solicit a DIO from their neighbours.

RPL uses a Trickle algorithm that regulates DIO control messages' transmission rate according to the current network conditions [16]. Trickle increases the transmission rate when a change in routing information is detected (i.e., an inconsistency) to update the network rapidly with new information [16]. In a steady case, Trickle exponentially reduces the transmission rate to limit the number of transmissions when there is no update to propagate. On the other hand, Trickle maintains a suppression mechanism in which a node limits redundant transmissions. Hence, the node suppresses the scheduled control packets if it detects that enough of its neighbours have transmitted the same piece of information [16].

1.2. RPL Security

The current RPL specification includes a few self-healing mechanisms, like loop detection and avoidance, global and local repair mechanisms. Furthermore, it defines security features like cryptographic security modes that are presented in the following subsections.

1.2.1. Self-healing Mechanisms

Loop Detection and Avoidance. Data packets must be transmitted upward from a child to its parent, where the parent has a lower Rank value than its child. Hence, nodes could use the packet direction and Rank information conveyed in control messages to detect inconsistency between the packet's path and the Rank rule between the sender and receiver nodes and discover possible loops [10].

Global and Local Repairs. RPL provides global repair and local repair mechanisms to fix links and node failures, and detect loops and other inconsistencies. Global repair is instituted by incrementing the DODAG Version Number field within the DIO message. Only the BR (DODAG root) could trigger this mechanism. However, any non-root node that detects an inconsistency (e.g., loop or link failure) can start a local repair. The node should poison its routes by announcing a rank of INFINITE RANK. Thus, it detaches itself from the DODAG and then re-attaches to the DODAG as a new joining node using a DIS message [10]. Malicious nodes could exploit both global and local repairs to trigger specific attacks against RPL networks.

1.2.2. Security Features

The self-organising, self-healing, and resource-constrained, as well as unreliable links, limited physical security, and dynamic topology of RPL networks, expose them to various internal and external threats. The RFC 6550 [10] states that RPL could use link-layer security mechanisms when they are available to secure message transmission. Furthermore, the RPL specification defines the following optional cryptographic security modes that nodes within an RPL network can adopt to ensure communication security.

Unsecure Mode. In this mode, RPL control messages are transmitted without any additional security features [10]. In this case, RPL relies on other layer security primitives, such as the MAC layer, to satisfy the network's security requirements [10].

Pre-installed Security Mode. In this mode, the nodes have pre-installed keys to generate and process RPL secured messages [10].

Authenticated Security Mode. Like the pre-installed mode, the nodes have pre-installed keys; nevertheless, they may only use the keys to join the network as a leaf. A router that needs to enter an RPL network requires another key from an authentication authority [10].

Despite the modes mentioned above, RPL networks remain vulnerable to existing and newly designed threats that have been extensively studied in the literature. Precisely, Rank attacks, Neighbour attack, DAO attacks, DIS attack, Version number attack, Local repair attack, Hello Flooding attacks, DIS attacks, Selective forwarding attack, Sinkhole and Blackhole attacks, Wormhole attack, and Sybil and CloneID attacks [12] [17] [18].

1.3. Contribution

We first introduced a solution to mitigate the Multicast DIS attack (M-DIS) effect on the RPL protocol in real-time in our previous work [19]. In the present paper, we give more details on the implementation and the evaluation of our solution that we called RPL-MRC. We evaluate RPL-MRC and its efficiency compared to the unsecured version of RPL. RPL-MRC is implemented on each node within the RPL-based LLN. We also evaluate the proposed solution under mobility using comprehensive Cooja-Contiki [20] experiments. The obtained results demonstrated that RPL-MRC is very effective in reducing the effects of M-DIS attack. As a result, it upgrades the RPL network performance significantly with respect to control overhead, packet delivery and power consumption. The concept of RPL-MRC can be used to detect similar attacks that use Multicast philosophy.

1.4. Paper Organisation

The rest of the paper is organised as follows. Section 2 presents an overview of the M-DIS attack. Section 3 sketches related work on the DIS attacks effects and countermeasures. Section 4 introduces the proposed solution and gives details on the mitigation mechanisms. Section 5 evaluates the performance of RPL-LLNs under M-DIS attack under various scenarios and presents the results. Finally, Section 6 rises conclusions and gives future works.

2. The Multicast DIS Attack (M-DIS)

RPL is based on IPv6 Neighbour Discovery mechanism. It relies on Multicast operations to set up the network topology. As presented above, a node within an RPL network sends a DIS message to solicit DIO messages from neighbouring nodes and join the DODAG. The DIS transmission interval varies from one RPL's implementation to another. In RPL Cooja-Contiki simulator [20], it is handled using RPL_CONF_DIS_START_DELAY and RPL_CONF_DIS_INTERVAL constants. After a node starts (i.e., after booting), it delays the transmission of its first DIS message according to the RPL_CONF_DIS_START_DELAY value. A node aiming to join the network continuously transmits DIS messages within the RPL_CONF_DIS_INTERVAL fixed interval until it receives a DIO message from its neighbours. Upon receiving a DIO message, it stops transmitting DIS messages and joins the network by sending a DAO message to its selected parent.

LLNs are not tamper-resistant, and nodes do not have a significant security defence. Hence, an adversary can compromise some nodes, reprogram and redeploy them into the network. As a consequence, even in the case of a secure RPL (See Section 1.2.2), the compromised nodes can use the pre-configured group key [10], and can normally participate in the network operations, and thus trigger attacks. In the M-DIS attack, the attacker exploits the RPL features mentioned above and frequently sends multiple Multicast DIS messages to its neighbours. Upon receiving a Multicast DIS message, the neighbouring nodes reset their DIO (Trickle) timers to the minimal value defined in the RPL implementation (2^{12} seconds) and send Multicast DIO messages containing the up-to-date routing information [10], as illustrated in Figure 1.

3. Related Work

RPL routing attacks have been widely studied in the literature, and many countering solutions have been proposed [17] [18] [21] [22]. This section focuses on the works around DIS attack, also known as Hello Flooding attack. Currently, there are a few works only to mitigate DIS attack.

The authors in [23] proposed a lightweight specification-based intrusion detection system (IDS) to detect the neighbour and DIS attacks. The IDS uses Extended Finite State Machine to define a profile of RPL (i.e., normal behaviour) and detect anomalies. The authors partitioned the network into clusters. Each cluster head requests its members to record a set of topology information periodically and report it, such as the DIS sequence and the number of received DIS. If the number of DIS messages received from a node is more than a threshold, the node is considered as

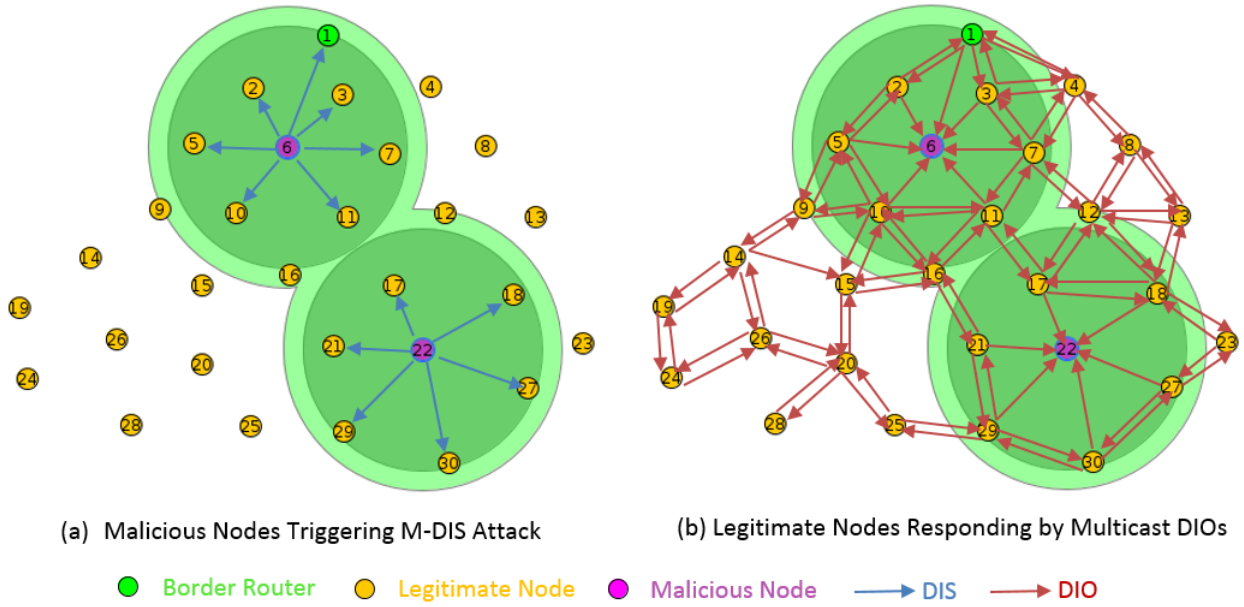

Figure 1: Multicast DIS Attack Illustration

an attacker. This solution introduces communication overhead and is less accurate when it works for a long time. Besides, it is designed for static networks.

The authors in [24] conducted extensive simulation experiments to evaluate the performance of RPL under DIS attack. According to their results, the attack significantly increases energy consumption and decreases the node lifetime. They concluded that the DIS attack is an extremely severe Denial of Service (DoS) attack for RPL-based LLNs. In addition, the authors in [25] implemented the DIS attack. They demonstrated that this attack negatively impacts the usage of nodes' resources with a decrease of 2% in LPM and an increase of 226%, 1275%, 81%, and 171% in the CPU Time, TX (transmitting) Time, RX (receiving) Time, and battery consumption, respectively.

As a follow-up to their work, the authors proposed a hybrid threshold-based IDS that uses the packet rate (i.e., DIS message sending rate) and the packet interval to detect the attack. In addition, the nodes' traffic is forwarded to the border router that will decide on the stat of a node (i.e., malicious or not) [26]. The solution in unstable as the results depend on the number of detectors within the network. Furthermore, it introduces communication overhead and is designed for static networks.

An anomaly-based lightweight IDS using threshold values has been proposed for dealing with the neighbour and DIS attacks in [27]. Unlike the work cited above [23], the IDS is fully distributed in every node of the RPL network where each node monitors its neighbours to detect the attacks. The authors defined a profile of normal behaviour for networks with different sizes (i.e., 20, 30, and 40 nodes). Every node stores the number of DIS messages received from its neighbours at specific time intervals. Afterwards, the maximum number of DIS messages received in all networks is calculated as the threshold to use. If the number of DIS messages received from a neighbour at specific time intervals is more than the threshold, that neighbour is considered as an attacker. The proposed IDS deals with the DIS attack within a static network only.

The authors in [28] proposed a solution, named Secure-RPL, to mitigate the effect of DIS flooding attacks. Secure-RPL approach suggests discarding all DIS messages received before the expiry of the RPL_CONF_DIS_INTERVAL from a particular neighbour. Many Information such as sender IP address, previous DIS message receiving time, and the total number of DIS messages received since the last reset are collected and used to detect the intruder. This solution has several drawbacks. A Sybil attacker can use different identities to divert the mitigation mechanism. Moreover, even though all non-attacker nodes are configured with the same DIS interval, they need to be synchronised to detect the DIS attack (in which malicious node sends DIS after the expiry of DIS interval). Furthermore, the authors tested their solution for small networks of 8 and 16 nodes with one attacker.

In another work [29], the authors examined the DIS attack's effects on energy efficiency and the DODAG construction. They demonstrated with simulations that the malicious node's neighbours are highly affected by the attack in terms of power consumption, then the nodes present at extreme boundaries. Indeed, the interference increases for all nodes with the presence of a malicious node. Accordingly, the ON and transmission periods increase, especially for the neighbours of the malicious node. Furthermore, they concluded that the attack affects the DODAG construction in the malicious node's transmission range.

In the field of machine learning-based IDSs, a compression header analyser based IDS (CHA-IDS) to detect Hello Flood, Sinkhole, and Wormhole attacks in an RPL network has been introduced [30]. The authors generated a dataset of 77 features and compared MLP, SVM, J48, NB, Logistic, and RF classifiers. The results showed that J48 performed better than the other classifiers for that specific configuration. Even though this approach presents a good background for IoT ML-based IDS, the authors considered one topology and a small network of eight nodes. In [31], the authors applied a deep-learning approach with five hidden layers to detect RPL routing attacks. The authors generated datasets for decreased Rank, Hello Flooding, and version number attacks relaying on different topologies. The obtained performance results in terms of F1-score for each dataset have been 94.7%, 99%, and 95%, respectively.

One drawback for the solutions that use a threshold parameter to detect the DIS attack is how to set a threshold for different configurations and topologies, especially for a dynamic network? In this case, we suppose that machine learning-based solutions are more appropriate. The second one is that several solutions assume that the attack is triggered after the DODAG stability; however, an attacker can start the attack before the setup of the DODAG like a zero-day attack. While the nodes count the number of DIS messages to compare it with a threshold, the malicious nodes affect the performance of the network, which is another disadvantage for such solutions. Otherwise, the detection time (related to the counting and the threshold) will be higher with the growing size and the network's dynamicity, which negatively influences the network's performance.

4. The Proposed Approach: RPL-MRC

Two complementary mitigation mechanisms have been proposed to address the M-DIS attack: Response Delay and Timer Readjustment. We integrated Response Delay into the dis_input function of the RPL implementation, whereas the Timer Readjustment has been integrated into the new_dio_interval function responsible for the timer's reset.

4.1. Response Delay

In the present work, we propose to reduce the impact of the M-DIS attack on RPL-based LLNs. In RPL-MRC, RPL itself is adapted to reduce the response to Multicast messages. M-IDS is inspired by the Multicast Listener Queries (LMQ) principle described in the RFC 3810 [32]. Multicast routers send MLQ massages in Querier State to query the multicast listening state of neighbouring interfaces. In the Queries format, a two-bytes field named the Maximum Response Code (MRC) specifies the maximum time allowed before sending a responding Report. It represents a floating-point value. The actual permitted time to respond is called the Maximum Response Delay (MRD). MRD is expressed in units of milliseconds and is derived from the MRC.

As the MLQ massages presented in RFC 3810, RPL-MRC uses a Maximum Response Code (MRC) field to reduce responses to DIS messages Multicast. To this end, we redefined the DIO Base Object as follows. We use the one-byte Reserved field as an MRC field set by the border router, as depicted in Figure 2. The MRC value must be greater than the I_{min} value of the Trickle timer and smaller than the I_{max} value (i.e., I_{min} plus the redundancy value k), as defined in Section 4.2. On receiving a Multicast DIS message, instead of responding immediately with a Multicast DIO message, the legitimate node delays its response by a random amount of time in the range [MRD/2, MRD], where the MRD value is calculated as in equation 1. In addition, we restricted the number of Multicast responses as follows. While delaying the response, every node tracks the number of DIO messages responding to the DIS Multicast. Suppose their number exceeds a pre-specified threshold less than the one allowed by the Trickle timer (i.e., the redundancy variable defined in Section 4.2). In that case, the node cancels its pre-programmed response.

$$\text{MRD} = \begin{cases} 2^{MRC} & \text{if } I_{min} < \text{MRC} < I_{min} + k, \text{ and k} > 3 \\ 2^{I_{min}+3} & \text{else.} \end{cases} \quad (1)$$

4.2. Timer Readjustment

The Trickle algorithm involves three configuration parameters: i) the maximum interval size (I_{max}); ii) the

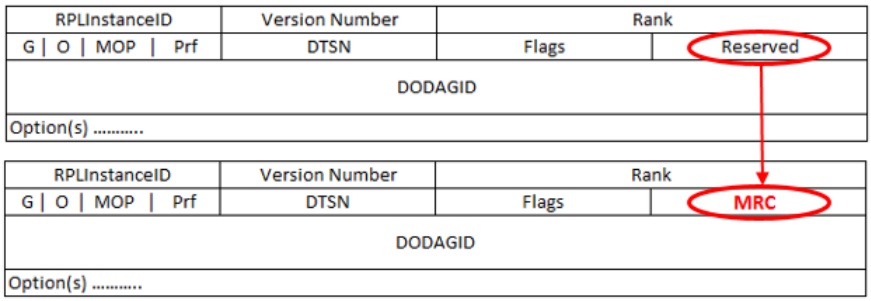

Figure 2: New DIO Message

minimum interval size (I_{min}); iii) the redundancy constant (k). Furthermore, it maintains three variables: 1) the size of the current interval (I); 2) a counter (c); 3) a specific time within the current interval (t) [16]. Each node is responsible for handling its own interval. The interval boundaries are [I_{min}, I_{max}]. This interval is divided into sub-intervals (periods). The first sub-interval starts with $I_{start} = I_{min}$ and ends with $I_{end} = I_{start} * 2$. Each time the first sub-interval is finished, a new sub-interval starts until reaching the end of the primary interval (i.e., I_{max}) [16], as illustrated in Figure 3. In the case of RPL, whenever a node hears a consistent DIO transmission from its neighbours, it increments the counter 'c'. At time t, the node transmits Multicast DIO if the counter 'c' is less than the redundancy constant 'k'. If not, the node suppresses the scheduled DIO transmission, waits until the current sub-interval 'I' has expired, and then doubles the sub-interval length [16]. Each time the node needs to check if it reaches the maximum of the interval I_{max} [16]. When a node receives a Multicast DIS

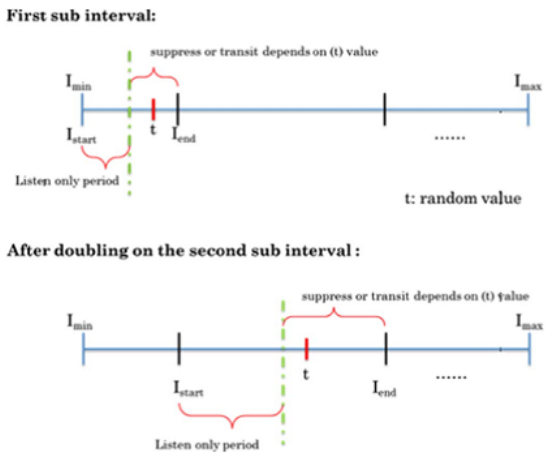

Figure 3: The Trickle Algorithm for a Node [33]

message, and 'I' is greater than I_{min}, it terminates (suppresses) the scheduled transmission of DIO messages (i.e., at the current sub-interval). It reinitialises the DIO Trickle timer from a sub-interval of a minimum length (i.e., sets I to I_{max}), as shown in Figure 4. If 'I' is equal to I_{min} when the node hears the Multicast DIS transmission, it does nothing (i.e., it waits for the scheduled DIO at time t) [16]. In RPL-MRC approach, the node reinitialises the Trickle timer following the MRD value, as in Equation 2. Indeed, the aim is to reduce the number of exchanged DIO messages and stabilise the network.

$$\text{Timer} = \begin{cases} \text{Reset to MRD} & \text{if MRD} < \text{Current-time} \\ \text{Do not reset} & \text{else.} \end{cases} \quad (2)$$

The pseudocode in Algorithm 1 summarises the proposed approach.

Algorithm 1 HelloFlooding and DIS Attacks Prevention

Require: MRC
  Calculate Maximum Response Delay (MRD) using MRC as defined in Equation 1
  if a node receives a multicast DIS message **then**
    It delays the response (sending a Multicast DIO) by a random amount of time in the range [MRD/2, MRD]
    if the number of response from its neighbours reaches the threshold defined by the border router **then**
      It cancels the pre-programmed response
    else
      once the delay has expired, it reinitialises the trickle timer and sends back a DIO
    end if
  end if

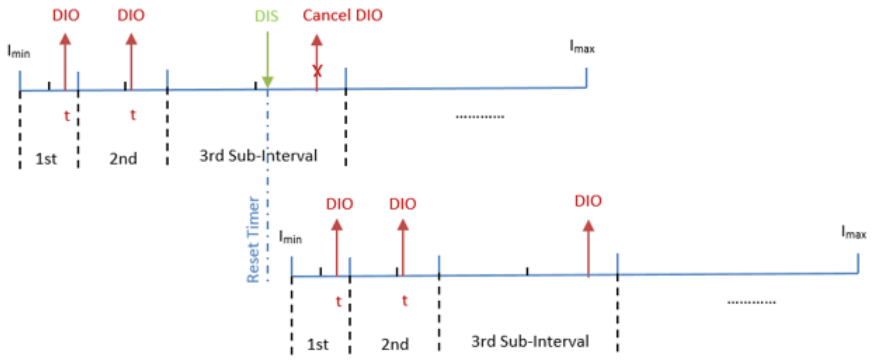

Figure 4: The Trickle Timer on Receiving Multicast DIS Message

5. Approach Evaluation

5.1. Performance metrics

As presented in the literature [23] [24] [29], the DIS attack influences significantly the control overhead, especially the number of DIO messages and energy consumption. Hence, the performance metrics used to evaluate RPL-MRC are as follows.

- Control packer overhead: It is the total number of DIO messages transmitted during the simulation.
- Power consumption (mW): It is the average power consumed by all the nodes in the network during the simulation. The calculation of the power consumption of each node is done by adding up the energy consumed on CPU (listening state), LPM (low power idle state), RX (radio listen state), and TX (radio transmit state).

In addition to the control overhead and power consumption metrics, we evaluated the data packets overhead.

- Data packets overhead: It is the total number of data packets received by the border router during the simulation. We also recorded the number of duplicated data packets to highlight the instability of the network.

5.2. Simulation Settings

Using the Cooja-Contiki simulator, we simulated three topologies of 30 nodes each, with one border router. We set up three main scenarios: (1) RPL without attack, (2) RPL under attack, and (3) RPL with RPL-MRC. We implemented sub-scenarios, where we varied the number of attackers. We also varied the frequency of the attack and the value of MRC. We used Tmote sky nodes and the radio protocol UDGM (Unit Disk Graph Radio Medium) with distance loss as a link failure model as it provides a real-world emulation of the lossy links and shared media collision among RPL's nodes. Additionally, we used the CSMA/CA for the link layer and the ContikiMAC as the radio duty cycling (RDC) protocol.

We run every simulation for 15 minutes. The nodes were distributed in an area of 300m x 250m. We used the RPL-collect package for packets generation, where each node sends one packet of 46 bytes every 60 seconds. For the performance metrics, we used radio messages and collect-view tools. Five runs were conducted for each scenario, and values were averaged. Besides, the proposed solution has been evaluated for the SybM attack defined in [19]. Table 1 summarises the parameters used for the simulations.

Table 1: Simulation Parameters

Parameter	**Value**
Simulator	Cooja-Contiki 3.0
Simulation time (mn)	15
Network area	300x250m2
Node type	Tmote Sky (telosB)
Number of nodes	30
Number of malicious nodes	2, 5, 10
Attack frequency (s)	3, 6, 10, 15, 30
MRC Values	13, 14, 15, 16, 17
Transmission range	70m
Interference range	60m
TX, RX	70%, 100%
MAC	ContikiMAC
Link failure model	UDGM with Distance Loss
Traffic rate	One packet sent every 10 seconds
Packet size	46 bytes

5.3. The Effect of the M-DIS Attack Frequency

To evaluate the effect of the attack frequency on the RPL performance, five malicious nodes were distributed uniformly in the network to ensure covering the vast majority of benign nodes and maximise the network's damage. MRC is set to 15 (which is equivalent to MRD equal to 32,768 seconds). We selected MRC=15 because it gives the best results as it can be seen in Section 5.5. The attack is triggered in intervals of 3, 6, 10, 25, and 30 seconds. Native RPL (RPL), RPL under M-DIS attack (RPL-DIS), and RPL under M-DIS attack with MRC countermeasure (RPL-MRC) were evaluated in terms of the metrics in Section 5.1.

Control Overhead. Figure 5 shows the performance of the network in terms of DIO messages overhead following different attacking intervals. It is noticed that the number of DIO messages sent in RPL-DIS scenario is very high compared to the native RPL and RPL-MRC regardless of the frequency of the attack. Nevertheless, we can observe that in RPL-MRC scenario, the DIO overhead has been decreased by 86%, 84%, 84%, 81%, and 79% for 3, 6, 10, 15, and 30 seconds intervals, when compared to RPL-DIS scenario. Indeed, RPL-MRC has performed very well, reducing the overhead to almost the one generated in the native, RPL. This is because RPL-MRC is executed every time a Multicast DIS message is received, even from legitimate nodes. We observe that in some cases (e.g., 10s and 30s intervals) the overhead is lower than native RPL. This could be because the nodes did not reset their timers according to the rule in equation 2 (i.e., the current time is less than MRD), in addition to the execution of RPL-MRC that reduces DIOs response to legitimate DIS Multicast.

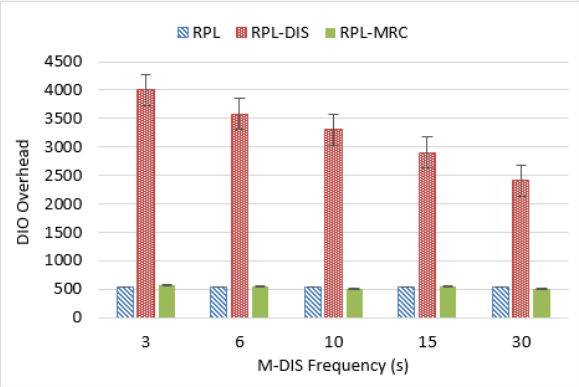

Figure 5: Control Overhead Vs Attack Frequency.

Power Consumption. As observed in Figure 6, the RPL-DIS network suffers heavily in terms of power consumption due to the attackers being able to flood the network with many DIS and DIO messages. However, following the RPL-MRC mitigation mechanism, the average power consumption has been reduced by 53%, 48.5%, 48%, 41%, and 34% for attack frequency of 3, 6, 10, 15, and 30 seconds, respectively. Indeed, the decline in the number of transmitted DIOs has resulted in lower power consumption. Both results (i.e., control overhead and power consumption) are justified by executing the RPL-MRC mechanism that redefines how to respond to a DIS Multicast.

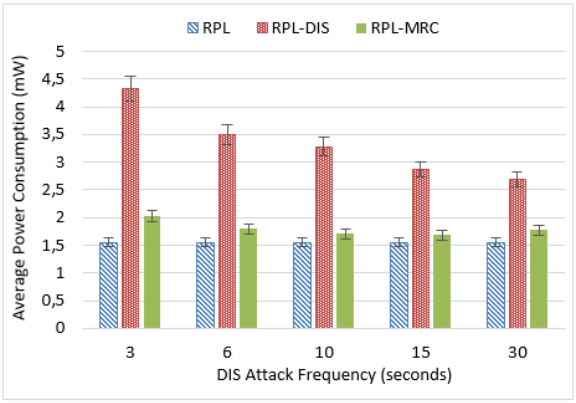

Figure 6: Power Consumption Vs Attack Frequency.

Data Packets Overhead. The results in Figure 7 demonstrate that the data packets overhead increases under the M-DIS attack. This is due to the increase of DIO overhead, which suppresses communication channel availability, forms a locally unstable network, and thus induces generating duplicate data packets. We notice that both the number of duplicate data packets and the number of delivered packets have been reduced using RPL-MRC countermeasure. In fact, under RPL-MRC, the network is more stable because the DIO overhead is reduced significantly.

5.4. The Effect of the Number of Attackers

To evaluate how RPL-MDS performs according to the number of attackers present in the network, we implemented the M-DIS attack with different numbers of malicious nodes (2, 5, and 10) and an MRC set to 15.

Control Overhead. Figure 8 shows the impact of varying the number of malicious nodes on the amount of control message exchanged in the network. In RPL-DIS

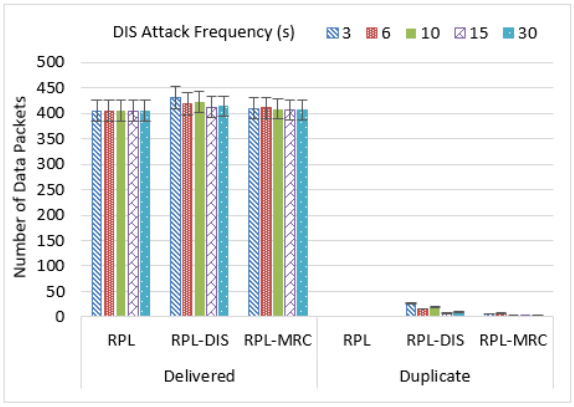

Figure 7: Delivered and Duplicate Data Packets Vs Attack Frequency.

scenario, the control overhead increases when the number of attackers grows because the attackers were distributed uniformly in the network. Thus, a large number of legitimate nodes are infected by the attack. In RPL-DIS, all nodes in a malicious node's radio range reset their Trickle timers every time they receive a DIS Multicast, and hence, send frequently DIO messages that are propagated in the network. However, RPL-MRC mechanism regulates the reset of the Trickle timer and the transmission of DIO messages in a way to reduce the overhead in the network. As a result, the overhead was decreased by 79%, 98.7%, and 90% in the presence of 2, 5, and 10 attackers, when compared to RPL-DIS.

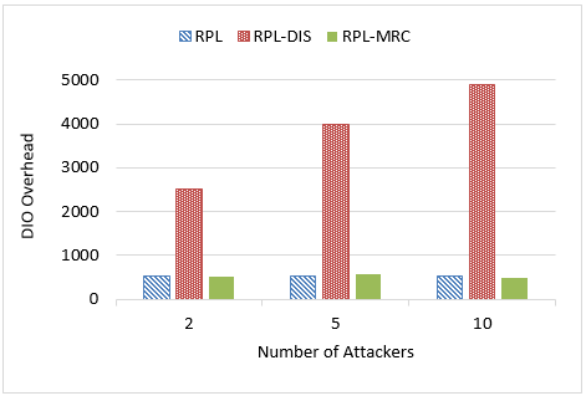

Figure 8: Control Overhead Vs. Number of Attackers.

Power Consumption. By analysing Figure 9, we realise that as the number of malicious nodes increases, the energy consumption increases significantly in the RPL-DIS scenario. Considering that more attackers exist in the network, more legitimate nodes respond to the attack by Multicast more DIO messages, resulting in larger power consumption. Albeit the energy consumption under RPL-MRC scenario is more than the native network (RPL scenario), it remains very good compared to the network under attack (RPL-DIS scenario). RPL-MRC mechanism was able to decrease the control overhead for a different number of attackers and, consequently, the network's overall power consumption. Although the number of DIOs has decreased, we notice that the energy increases as the number of attackers increases. Indeed, malicious nodes consume more energy on transmitting DIS messages (frequency of 3 seconds per attacker), which means that the network's average power consumption increases.

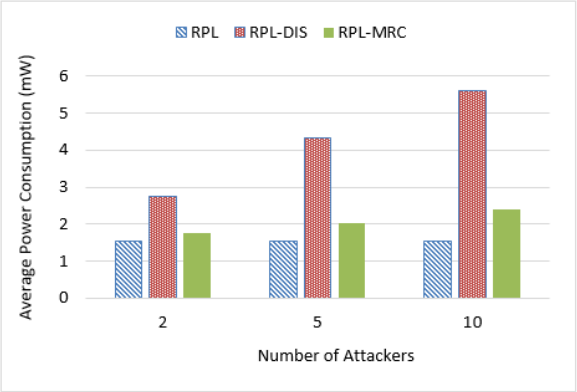

Figure 9: Power Consumption Vs Number of Attackers

Data Packets Overhead. Under M-DIS attack (for both RPL-DIS and RPL-MRC scenarios), the border router receives a larger number of original and duplicate data packets, as shown in Figure 10. When the node does not receive the acknowledgement, it schedules retransmission, leading to a duplicate packet. It is evident that data packets may be correctly received, and the corresponding acknowledgement may be lost or even may collide due to transmissions unreliability resulted from the increase of control overhead. However, regardless of the number of malicious nodes, RPL-MRC makes the network more stable. As a result, the number of duplicate packets is reduced.

5.5. The Effect of the MRC Parameter

This section investigates the MRC parameter setting's effect on the RPL network performance by increasing the MRC value, starting with 13 and incrementing it by one to a maximum of 17 (i.e., 13, 14, 15, 16, and 17). The MRC values correspond to MRD values of 2^{13}, 2^{14}, 2^{15}, 2^{16}, and 2^{17}, respectively).

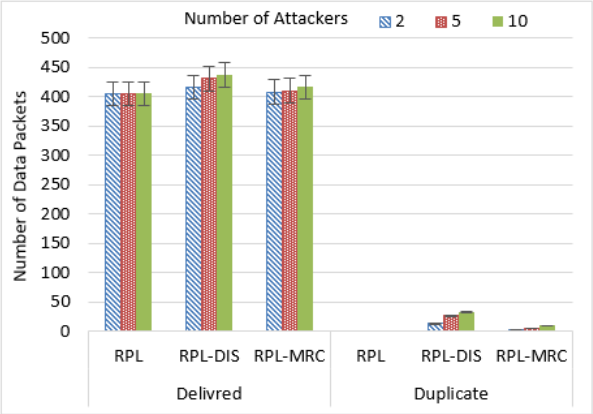

Figure 10: Delivered and Duplicate Data Packets Vs. Number of Attackers.

Control Overhead. It is clear from Figure 11 that RPL-MRC reduces the control overhead significantly, whatever the MRC value. Indeed, the control overhead has been decreased by 56%, 74%, and 86% for MRC equal to 13, 14, and 15, respectively. We notice that setting a small value for MRC getting closer (approximates) to the Trickle timer minimum interval (i.e., 13 and 14) induces more control overhead. Whereas, MRC values from 15 give approximately the same results and an overhead close to the native RPL one.

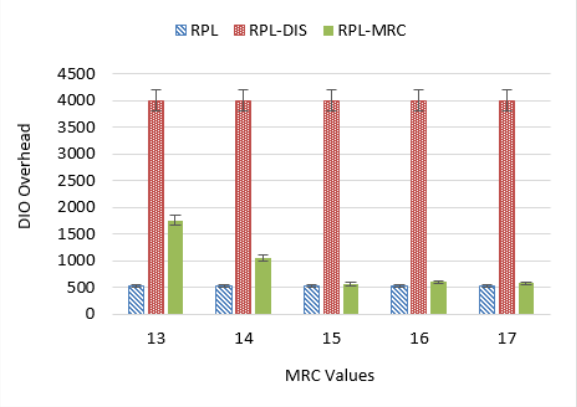

Figure 11: Number of DIO under Different MRC Values.

Power Consumption. Similarly to all of the above cases, the decrease in power consumption under RPL-MRC is a logical consequence of reducing control message overload. It is evident from Figure 12 that the MRC values from 15 give better results in terms of energy consumption. The following points can explain the results for both control messages overhead and energy consumption:

- The nodes reset their timers but suppress the delayed DIOs because the threshold of transmissions is reached.
- The nodes do not reset their timers because the current interval (period) is less than the response delay value. It could occur for MRC values of 15, 16, and 17.
- The nodes reset their timer to a value greater than the I_{min} defined by the Trickle timer.

As in Section 5.4, malicious nodes consume more energy on transmitting DIS messages, which implies an increase in the overall network's average power consumption. However, the results remain satisfactory with a decrease between 35% and 53% compared to RPL-DIS.

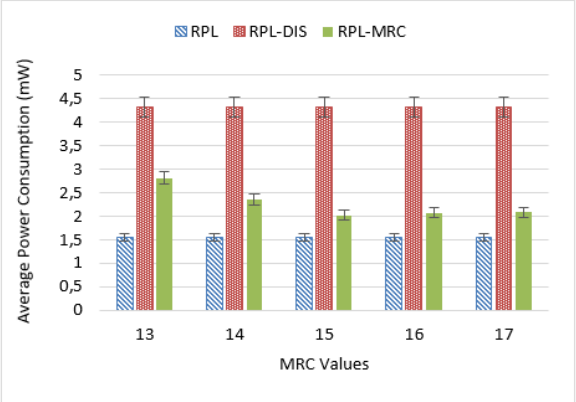

Figure 12: Power Consumption under Different MRC Values.

Data Packets Overhead. The results from Figure 13 have also demonstrated that the DIS attack may moderately affect data packets' delivery. RPL-MRC has improved the network's stability for all MRC values and consequently decreased the data packets overhead.

5.6. The Effect of Mobility

In our previous work [19], we introduced the SybM attack, which is a combination of Sybil and M-DIS attacks where malicious nodes are mobile. Figure 14 represents an illustrative example of the SybM attack model. Initially, each attacker is placed at a random location and sends data packets periodically to the border router as a legitimate node. Afterwards, the malicious nodes trigger the attack following a time interval.

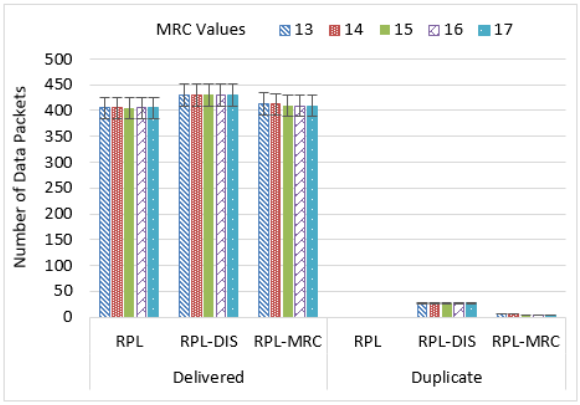

Figure 13: Delivered and Duplicate Data Packets for Different MRC Values.

Each adversary node involves a set of its Sybil identities alternately and periodically while moving through the network to the border router. Just after moving to a new location, the attackers send Multicast DIS messages to their new neighbours within the network using new Sybil identities. As a result, neighbourhood connectivity will change, and obviously, more DIO messages will be exchanged. We demonstrated with extensive simulations that the harmful effects of the SybM attack (dynamic case) on RPL networks performance surpass the impact of the DIS attack (static case) [19] vastly. In this section, we study the effect of the proposed solution (RPL-MRC) on RPL under SybM attack. We simulated a network of 50 TelosB nodes (Sky motes) with one border router and 49 senders. Table 2 highlights the simulation parameters specific for SybM attack.

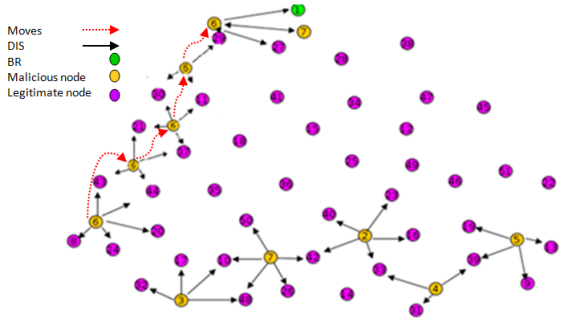

Figure 14: SybM Attack Illustration.

Control Overhead. Figure 15 demonstrates that the overhead generated in all scenarios exceeds the one generated in previous sections, even if the simulation duration is 5 minutes. In fact, we used a larger network of 50 nodes that generate more traffic to construct and maintain the RPL topology. As depicted in the figure, SybM attack caused an extra overhead of 55.7%, which is more than the double compared to the one generated from native RPL. Nonetheless, RPL-MRC behaves like in the static case (RPL-DIS) and reduces the attack's effect (RPL-SybM) on control overhead by 55%. In conclusion, RPL-MRC is very efficient to reduce the response to a DIS Multicast in a dynamic (mobile) network.

Table 2: Simulation Parameters for Mobility Scenario

Parameter	Value
Simulation time (s)	330
Network area	300x200m2
Number of nodes	50
Number of malicious nodes	10
Attack frequency (s)	3
Number of moves (identities)	5 per attacker
MRC Value	15

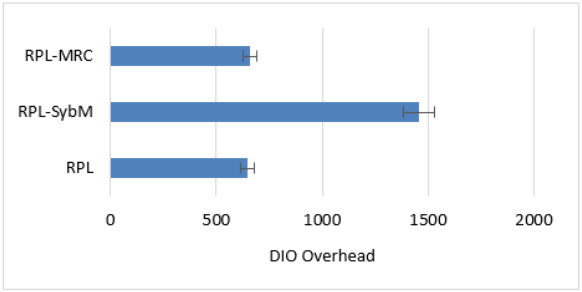

Figure 15: Control Overhead under SybM Attack.

Power Consumption. The power consumption increases with the size of the network following the increase in control overhead. The Figure 6, Figure 9, Figure 12, and Figure 16 reveal that the power consumption for native RPL with 50 nodes and 5 minutes simulation time increased by 21% compared to 30 nodes and 15 minutes simulation time. RPL-SybM generated an extra power consumption of 42%, and RPL-MRC reduced it by 33.6%. Hence, RPL-MRC additional overhead is about 8.4%, which is acceptable as the first line of defence.

Data Packets Overhead. As shown in Figure 17, in the presence of SybM attack, the duplicate data packets increase, hence increasing the delivered ones. This can be explained by the mobility of the attackers and the Multicast of DIS messages that render the network unstable.

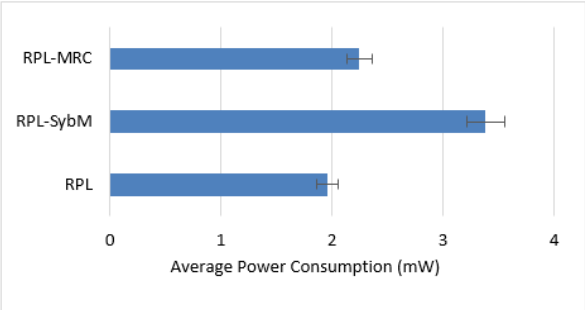

Figure 16: Power Consumption under SybM Attack.

RPL-MRC succeeds in reducing the number of duplicate packets by 91%.

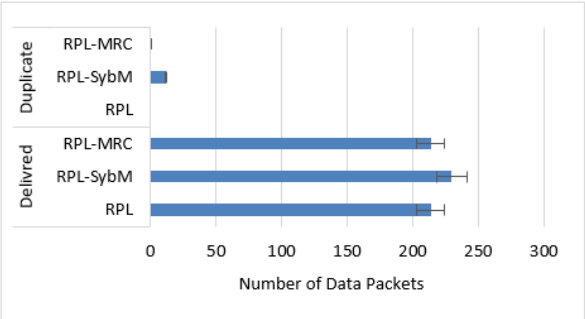

Figure 17: Data Packets Overhead under SybM Attack.

6. Conclusion

A node aiming to join the RPL DODAG sends a DIS message to solicit topology information. On receiving a DIS Multicast, neighbouring nodes reset their trickle timer to its minimum value and replay by DIO Multicast. Indeed, a simple DIS Multicast can significantly increase the number of exchanged control messages. The abrupt increase of control overhead increases the overall power consumption of the network and further reduces the network lifetime. This study proposed a solution to deal with the DIS Multicast issue that an intruder can use to harm the network. The results highlighted the efficiency of RPL-MRC mechanism for reducing control overhead, power consumption, and data packet overhead. We studied the effect of the approach on different scenarios (e.g., varying the attack frequency, varying the number of attackers, varying the proposed parameter MRC, and under mobility). We concluded that RPL-MRC achieves high performance in all cases. We demonstrated the RPL-MRC scalability as it presented good fulfilment for a larger network. We suggest that our solution could be combined with other detection methods, such as the threshold-based to protect RPL-LLNs. Indeed, RPL-MRC can reduce the attack's effect before the attackers are detected and discarded from the network.

We intend to validate the proposed solution's efficiency in real testbeds and a fully dynamic network in our future work. Furthermore, it is interesting to explore the integration of RPL-MRC to an intrusion detection system.

7. Acknowledgements

This work is financially supported by the Research Centre on Scientific and Technical Information (CERIST).

References

[1] L. Atzori, A. Iera, G. Morabito, The internet of things: A survey, Computer Networks 54 (15) (2010) 2787 – 2805. doi:https://doi.org/10.1016/j.comnet.2010.05.010.

[2] S. Li, L. Da Xu, S. Zhao, The internet of things: a survey, Information Systems Frontiers 17 (2) (2015) 243–259.

[3] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, M. Ayyash, Internet of things: A survey on enabling technologies, protocols, and applications, IEEE Communications Surveys Tutorials 17 (4) (2015) 2347–2376. doi:10.1109/COMST.2015.2444095.

[4] J. Hui, P. Thubert, et al., Compression format for ipv6 datagrams over ieee 802.15. 4-based networks (2011).

[5] Routing over low power and lossy networks (roll). URL `https://datatracker.ietf.org/wg/roll/charter/`

[6] J. Hui, J. Vasseur, D. Culler, V. Manral, An ipv6 routing header for source routes with the routing protocol for low-power and lossy networks (rpl), Request for Comments 6554 (2012).

[7] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, P. Levis, Collection tree protocol, in: Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems, SenSys '09, ACM, 2009, p. 1–14. doi:10.1145/1644038.1644040.

[8] S. Dawson-Haggerty, A. Tavakoli, D. Culler, Hydro: A hybrid routing protocol for low-power and lossy networks, in: 2010 First IEEE International Conference on Smart Grid Communications, 2010, pp. 268–273. doi:10.1109/SMARTGRID.2010.5622053.

[9] J. Vasseur, N. Agarwal, J. Hui, Z. Shelby, P. Bertrand, C. Chauvenet, Rpl: The ip routing protocol designed for low power and lossy networks, Internet Protocol for Smart Objects (IPSO) Alliance 36 (2011).

[10] T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, J. Vasseur, R. Alexander, Rpl: Ipv6 routing protocol for low-power and lossy networks, RFC 6550, Internet Engineering Task Force (2012).

[11] T. Tsao, R. Alexander, M. Dohler, V. Daza, A. Lozano, M. Richardson, A security threat analysis for the routing protocol for low-power and lossy networks (rpls), RFC7416 (2015) 131.

[12] F. Medjek, D. Tandjaoui, I. Romdhani, N. Djedjig, Security threats in the internet of things: Rpl's attacks and countermeasures, in: Security and Privacy in Smart Sensor Networks,

IGI Global, 2018, pp. 147–178. doi:10.4018/978-1-5225-5736-4.ch008.

[13] J. Vasseur, M. Kim, K. Pister, N. Dejean, D. Barthel, Routing metrics used for path calculation in low power and lossy networks, RFC 6551, Internet Engineering Task Force (2012).

[14] P. Thubert, Objective function zero for the routing protocol for low-power and lossy networks (rpl), RFC 6552, Internet Engineering Task Force (2012).

[15] O. Gnawali, P. Levis, The minimum rank with hysteresis objective function, RFC 6719 (2012).

[16] P. Levis, T. Clausen, J. Hui, O. Gnawali, J. Ko, The trickle algorithm, Internet Engineering Task Force, RFC6206 (2011).

[17] A. Jain, S. Jain, A survey on miscellaneous attacks and countermeasures for rpl routing protocol in iot, in: A. Abraham, P. Dutta, J. K. Mandal, A. Bhattacharya, S. Dutta (Eds.), Emerging Technologies in Data Mining and Information Security, Springer Singapore, 2019, pp. 611–620.

[18] S. Choudhary, N. Kesswani, A Survey: Intrusion Detection Techniques for Internet of Things, International Journal of Information Security and Privacy (IJISP) 13 (1) (2019) 86–105.

[19] F. Medjek, D. Tandjaoui, I. Romdhani, N. Djedjig, Performance evaluation of rpl protocol under mobile sybil attacks, in: 2017 IEEE Trustcom/BigDataSE/ICESS, 2017, pp. 1049–1055. doi:10.1109/Trustcom/BigDataSE/ICESS.2017.351.

[20] F. Osterlind, A. Dunkels, J. Eriksson, N. Finne, T. Voigt, Cross-level sensor network simulation with cooja, in: Proceedings. 2006 31st IEEE Conference on Local Computer Networks, 2006, pp. 641–648. doi:10.1109/LCN.2006.322172.

[21] F. Medjek, D. Tandjaoui, I. Romdhani, N. Djedjig, A trust-based intrusion detection system for mobile rpl based networks, in: 2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), 2017, pp. 735–742. doi:10.1109/iThings-GreenCom-CPSCom-SmartData.2017.113.

[22] N. Djedjig, D. Tandjaoui, F. Medjek, I. Romdhani, Trust-aware and cooperative routing protocol for iot security, Journal of Information Security and Applications 52 (2020) 102467. doi:https://doi.org/10.1016/j.jisa.2020.102467.

[23] A. Le, J. Loo, K. K. Chai, M. Aiash, A specification-based ids for detecting attacks on rpl-based network topology, Information 7 (2) (2016) 25.

[24] C. Pu, Spam dis attack against routing protocol in the internet of things, in: 2019 International Conference on Computing, Networking and Communications (ICNC), 2019, pp. 73–77. doi:10.1109/ICCNC.2019.8685628.

[25] R. Smith, D. Palin, P. P. Ioulianou, V. G. Vassilakis, S. Shahandashti, Battery draining attacks against edge computing nodes in iot networks, Cyber-Physical Systems 6 (2020) 116 – 96.

[26] P. P. Ioulianou, V. G. Vassilakis, Denial-of-service attacks and countermeasures in the rpl-based internet of things, in: S. Katsikas, F. Cuppens, N. Cuppens, C. Lambrinoudakis, C. Kalloniatis, J. Mylopoulos, A. Antón, S. Gritzalis, F. Pallas, J. Pohle, A. Sasse, W. Meng, S. Furnell, J. Garcia-Alfaro (Eds.), Computer Security, Springer International Publishing, 2020, pp. 374–390.

[27] B. Farzaneh, M. A. Montazeri, 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. doi:10.1109/ICWR.2019.8765272.

[28] A. Verma, V. Ranga, Mitigation of dis flooding attacks in rpl-based 6lowpan networks, Trans. Emerg. Telecommun. Technol. 31 (2020).

[29] S. Sharma, V. K. Verma, Security explorations for routing attacks in low power networks on internet of things, The Journal of Supercomputing (2020) 1–35.

[30] M. N. Napiah, M. Y. I. B. Idris, R. Ramli, I. Ahmedy, Compression header analyzer intrusion detection system (cha-ids) for 6lowpan communication protocol, IEEE Access 6 (2018) 16623–16638. doi:10.1109/ACCESS.2018.2798626.

[31] F. Y. Yavuz, D. Ünal, E. Gül, Deep learning for detection of routing attacks in the internet of things, International Journal of Computational Intelligence Systems 12 (1) (2018) 39–58. doi:10.2991/ijcis.2018.25905181.

[32] R. Vida, L. Costa, Rfc 3810, Multicast Listener Discovery Version 2 (2004).

[33] M. B. Yassein, S. Aljawarneh, E. Masa'deh, A new elastic trickle timer algorithm for internet of things, Journal of Network and Computer Applications 89 (2017) 38 – 47, emerging Services for Internet of Things (IoT). doi:https://doi.org/10.1016/j.jnca.2017.01.024.