# Energy Depletion Attack in Low Power and Lossy Networks: Analysis and Defenses

Cong Pu Bryan Groves

***Abstract*—With increasingly popular computing devices endowed with sensing and communicating capabilities, low power and lossy networks (LLNs) are rapidly emerging as an important part of ubiquitous computing and communication infrastructure. In order to support the vision of Internet-of-Things (IoT) and its applications, a novel routing protocol for low power and lossy networks, also referred to as *RPL*, has been proposed to provide efficient and reliable communication and enable the integration of resource-constrained devices into the Internet. However, due to the shared wireless medium, the lack of physical protection, and instinctive resource constraints, RPL-based LLNs are undeniably vulnerable to various Denial-of-Service (DoS) attacks. In this paper, we propose a misbehavior-aware detection scheme, called *MAD*, against energy depletion attack in RPL-based LLNs, where a malicious node intentionally generates and sends a large number of packets to legitimate node to excessively consume the energy resource of intermediate nodes located along the forwarding path, and finally makes the resource-constrained network suffer from denial of service. In the *MAD*, each node maintains a count of the number of received packets from its child node within a specific time window, and then compares the count with a dynamically calculated threshold to detect potential energy depletion attack. We conduct extensive simulation experiments for performance evaluation and comparison with the original RPL with and without adversary, respectively. The simulation results show that the proposed scheme is a viable approach against energy depletion attack in RPL-based LLNs.**

***Index Terms*—Energy Depletion Attack, Denial-of-Service, RPL, Low Power and Lossy Networks, Internet-of-Things**

## I. Introduction

A rapidly growing pervasiveness and ubiquity of small and cheap computing devices (later nodes) endowed with sensing and communicating capabilities is leading the emergence of Internet-of-Things (IoT), and paving the way to the realization of IoT applications [1]. By 2019, the global IoT market is forecast to be valued at more than 1.7 trillion U.S. dollars, with the number of connected devices worldwide forecast to reach 20.35 billion in the same year [2]. With a variety of communication solutions such as Bluetooth, WiFi, 4G/5G, and ZigBee, and the recent advance in embedded and wireless devices, we envision that wirelessly connected smart nodes in the realm of IoT will enhance information accessibility and availability as well as improve our lives further.

As a major building block of promptly emerging IoT, low power and lossy networks (LLNs) comprised largely of resource-constrained nodes with the limited communication, computation, memory and energy are playing an indispensable role in creating an ubiquitous computing and communication environment. In order to bridge the gap between resource-constrained nodes and the IP world, routing protocols have been considered as one of key issues that is worth of investigation. A routing protocol, called Hydro [3], has been proposed for LLNs with the consideration of robust collection, point-to-point communication, and low footprint. However, this proprietary solution was a flash in the pan and did not attract enough attention. With the increasing demand of sharing information and knowledge as well as coordinating decisions, the Internet Engineering Task Force (IETF) Working Group has proposed a novel routing protocol for low power and lossy networks, also referred to as RPL [4], as the main candidate routing protocol for resource-constrained nodes in LLNs.

However, due to the shared wireless medium, and the lack of resources, physical protections, and security requirements of network protocol, RPL-based LLNs are admittedly vulnerable to various Denial-of-Service attacks [5]. Link-layer security mechanisms, such as IEEE 802.15.4 AES-128 [6] and Cisco's CG-Mesh [7], are recommended to protect RPL routing control messages and topology from external attackers who have no access to cryptographic materials [8]. But, link-layer security mechanisms are incapable of detecting an inside attacker who physically compromises a legitimate node, gains access to all stored information including public and private keys, and reprograms it to behave maliciously [9]. In addition, by reason of resource consumption, current RPL implementations are not willing to employ extra security operations, which greatly affect the performance of resource-constrained nodes [10], [11]. A security threat analysis of LLNs presented in [12] only discusses the well-known attacks targeting authentication, confidentiality, integrity, and availability, and suggests basic countermeasures. To sum up, this leaves RPL-based LLNs vulnerable and open to new Denial-of-Service attacks.

In this paper, we investigate an energy depletion attack and propose a corresponding countermeasure in RPL-based LLNs, where a malicious node intentionally generates and sends a large number of packets to legitimate node to excessively consume the energy resource of intermediate nodes located along the forwarding path, and finally causes denial of service in resource-constrained networks. The energy depletion attack primarily targets the vulnerability of point-to-point (P2P) routing mechanism in RPL by violating an implicit assumption, i.e., all intermediate nodes unhesitatingly and faithfully route all the received packets to destination node. Unlike the false

Cong Pu and Bryan Groves are with the Weisberg Division of Computer Science, Marshall University, Huntington, WV 25755, Email: {puc, groves54}@marshall.edu

978-1-7281-2080-5/19/$31.00 ©2019 IEEE
DOI 10.1109/ICDIS.2019.00010

data injection attack [13], where a malicious node periodically injects false data packets to mislead the system make wrong decisions, it is a non-trivial problem to detect the attack packets from normal data traffic in energy depletion attack. In light of these, we propose a light-weight countermeasure and its corresponding techniques to detect the energy depletion attack in RPL-based LLNs. Our major contribution is briefly summarized in the following:

1) We present and analyze the energy depletion attack with a preliminary result in RPL-based LLNs. This is the first in-depth work to investigate the performance impact of energy depletion attack in RPL-based LLNs.
2) We propose a misbehavior-aware detection scheme, called *MAD*, to efficiently detect and mitigate the energy depletion attack in RPL-based LLNs. In the *MAD*, each node maintains a count of the number of received packets from its child node within a specific time window, and then compares the count with a dynamically calculated threshold to detect potential malicious node.
3) We revisit and implement the original RPL with and without adversary for performance comparison. In addition, the original RPL without adversary is used as the lower bound of energy consumption and the upper bound of packet delivery ratio.

We develop a customized discrete event-driven simulation framework by using OMNeT++ [14] and evaluate its performance through extensive simulation experiments in terms of detection rate, detection latency, node behavior distribution, energy consumption, and packet delivery ratio. The simulation results indicate that the proposed countermeasure is a viable approach against energy depletion attack in RPL-based LLNs.

The remainder of the paper is organized as follows. The prior approaches are summarized and analyzed in Section II. The basic RPL operations and its potential vulnerability are summarized and analyzed with a preliminary result in Section III. The proposed countermeasure and performance evaluation with extensive simulation experiments are provided in Sections IV and V, respectively. We further explore the potential extensions of proposed countermeasure in Section VI. Finally, we conclude the paper in Section VII.

## II. Related Work

Over the past several years, the security of RPL has received a significant amount of research attention along with the emergence of Internet-of-Things and its applications. In [15], DIO suppression attack is investigated in IPv6-based wireless sensor and actuator networks. In the Dodge-Jam [16], a light-weight anti-jamming technique suitable for LLN is proposed to address the stealthy jamming attacks with small overhead. To detect forwarding misbehavior in LLNs running with RPL, a monitor-based approach is proposed in [17], where each node monitors the forwarding behaviors of the preferred parent node to detect the forwarding misbehaviors. The [18] proposes a dynamic threshold mechanism to mitigate DAO inconsistency attack in RPL-based LLNs, where a malicious node intentionally drops the received data packet and replies the forwarding error packet to cause the parent node to discard valid downward routes in the routing table. A new type of Denial-of-Service attack in LLNs, also referred to as hatchetman attack, is identified and investigated in [19]. In our previous work [20], the energy depletion attack is investigated with a preliminary result, indicating that the energy depletion attack is an extremely severe attack in resource-constrained networks. The [21] and [22] propose a heuristic-based detection scheme against the suppression attack in multicast protocol for LLNs, where a malicious node multicasts a series of spoof data messages with continuous sequence numbers to prevent normal nodes from accepting valid data messages and cause them to delete cached data messages. The [23] presents and investigates a new type of Denial-of-Service attack, called spam DIS attack, in RPL-based LLNs. The history of research efforts in RPL-based LLNs and future research directions on which RPL should evolve have been discussed in [8].

In addition, many researchers have explicitly studied the security issues in the Internet-of-Things. The [24] proposes a novel intrusion detection system, also called SVELTE, to secure IPv6 over Low-Power Wireless Personal Area Network (6LoWPAN) running with RPL from network layer and routing attacks. The [25] suggests two complementary and lightweight defense mechanisms to counter fragmentation attack in the adaptation layer of 6LoWPAN. The security capability of IEEE 802.15.4 MAC protocol as well as the limitations thereof in the context of Internet-of-Things are analyzed in [26]. Two detailed surveys of DoS attacks in the IoT can be found in [27], [28].

While the development of security in LLNs is still in the initial stage, a great deal of research effort has aimed to investigate a variety of attacks and countermeasures in similar environments. A camouflage-based detection scheme, also referred to as CAM, is proposed to detect the forwarding misbehavior in energy harvesting motivated networks (EHNets) in [29]. The EYES [30] is an extended version of the CAM. The [31] proposes an acknowledgment-based approach against stealthy collision attack in EHNets, where two malicious nodes coordinate their packet transmissions simultaneously to create the packet collision at a legitimate node. In [32], a single checkpoint-assisted approach integrated with timeout and hop-by-hop retransmission techniques is proposed to detect the selective forwarding attack in wireless sensor networks, where single or multiple malicious nodes randomly or strategically drop any incoming packet. In [33], a DSR-based bait detection scheme incorporated with a digital signature technique is proposed to detect routing misbehaviors in mobile ad hoc networks, where malicious nodes falsely claim a fake shortest route to a destination node to attract network traffic on purpose. The [34] proposes a jamming-resilient multipath routing protocol, also called JarmRout, so that intentional jamming and disruption, or isolated and localized failures do not interrupt the overall network performance of Flying Ad Hoc Networks. In [35], security vulnerabilities and threats imposed by the inherent open nature of wireless communications are examined. In order to improve the wireless network security among different

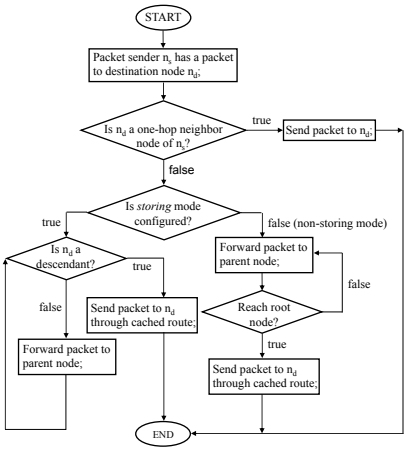



Fig. 1. An information flow of P2P routing mechanism.

layers, a variety of efficient defense mechanisms are presented.

In summary, various attacks and diverse countermeasures have been well studied in different networks and environments. However, little attention has been paid to energy depletion attack targeting the vulnerability of point-to-point routing mechanism in RPL-based LLNs.

## III. The RPL Routing Protocol and Energy Depletion Attack

In this section, we first briefly review the basic operations of RPL, and then present and analyze the energy depletion attack with a preliminary result.

### A. The RPL Routing Protocol

RPL [4] is a novel distance vector and source routing protocol designed for low power and lossy networks operating on IEEE 802.15.4 PHY and MAC layers. In order to maintain the network state information and topology, RPL constructs one or more Destination Oriented Directed Acyclic Graphs (DODAGs) which are differentiated by RPL Instance ID, DODAG ID, and DODAG Version Number. Each DODAG is associated with a set of normal nodes and one DODAG root (i.e., base station or gateway node), where normal nodes generate and forward data traffic and DODAG root is responsible for collecting the data measured by normal nodes, controlling normal nodes, and bridging the DODAG with the Internet.

*1) DODAG Construction:* In order to construct a DODAG and build upward routes directed from other nodes to the DODAG root, the DODAG root will issue a DIO control message, which includes the DODAG root's ID, the rank of the DODAG root, and an Objective Function which describes the routing metrics and constraints. Any node that receives the DIO message and is willing to join the DODAG should add the DIO message sender to its parent list, compute its own rank according to the piggybacked Objective Function, and pass on the DIO message with the updated rank information. Here, the rank is used to imply the node's position relative to other nodes with respect to the DODAG root, and the rank of nodes along any upward route to the DODAG root should be monotonically decreasing to avoid any routing loop.

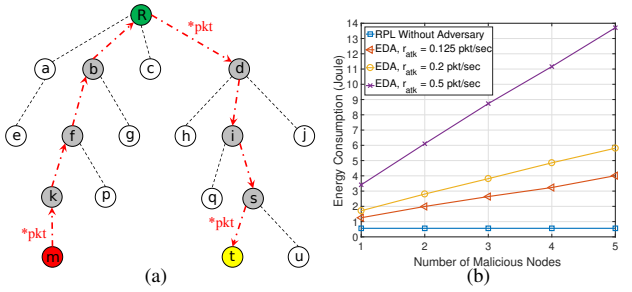



Fig. 2. An example of energy depletion attack and the performance impact of energy depletion attack: (a) A malicious node $n_m$ intentionally generates and sends a large number of packets $*pkt$ to a destination node $n_t$. Here, green node is the DODAG root, and red dash-dotted lines represent forwarding path. (b) The energy consumption against number of malicious nodes and attack rate ($r_{atk}$). Here, a 200×200 ($m^2$) network area is considered, where normal packet injection rate is 0.1 pkt/sec.

If a new node wants to join the existing network, it can request topology information from the neighbor nodes in the adjacent DODAGs by broadcasting a DIS control message. To build downward routes from the DODAG root to other nodes, the destination node needs to issue a DAO control message to propagate reverse route information and record the nodes visited along the upward routes. After passing the DAO message to the DODAG root, a complete downward route between the DODAG root and the destination node is established. Finally, the DODAG root replies a DAO-Ack message as a unicast packet to the source of DAO message.

*2) Point-to-Point Routing Mechanism:* RPL provides point-to-point (P2P) routing mechanism for any two nodes to communicate in the DODAG [36]. If the destination node is the one-hop neighbor node of the packet sender, the latter directly sends the packet to destination node without going through its parent node. Otherwise, the operations of P2P routing mechanism depend on whether RPL is configured as storing or non-storing mode. In the non-storing mode, except for DODAG root, each node does not store any routing information about downward nodes. In this case, any packet must be first delivered through the upward route to the DODAG root, which will forward the packet to destination node. In the storing mode, each node locally caches the routing information about downward nodes. If the destination node is a descendant of packet sender, it forwards the packet to destination node via cached downward route. Otherwise, the packet is forwarded to parent node, at which the same aforementioned operations will be applied to send the packet to its destination node. As such, the packet will be forwarded through upward routes until reaching the node that is the first ancestor of both packet sender and destination node. Here, an information flow of P2P routing mechanism is illustrated in Fig. 1.

### B. Energy Depletion Attack

Normally, P2P routing mechanism is used to initiate data transfer, send end-to-end acknowledgments, or carry out infrequent network diagnostics [3]. However, the vulnerability of P2P routing mechanism, e.g., all intermediate nodes unhesitatingly and faithfully route the received packets to destination node, can be exploited by adversary to attack the network as

well. Considering an example in Subfig. 2(a), a malicious node $n_m$ generates and sends a large number of packets, denoted as $^*pkt$, to a destination node $n_t$. In the non-storing mode, all packets first have to be forwarded through upward route to the DODAG root $n_R$, which forwards the $^*pkt$ to destination node $n_t$ according to cached downward route. In the storing mode, all packets will be forwarded through upward route until reaching the first common ancestor of packet sender $n_m$ and destination node $n_t$, which is the DODAG root $n_R$, and then delivered to destination node $n_t$. Thus, no matter which mode is configured, all intermediate nodes (i.e., $n_k$, $n_f$, $n_b$, $n_d$, $n_i$, and $n_s$) located along the forwarding path between packet sender $n_m$ and destination node $n_t$ have to receive and forward a large number of packets, which consume a significant amount of energy resource. In LLNs, since each node is equipped with a limited amount of energy, energy depletion attack can easily deplete the limited energy resource of legitimate nodes, and finally makes the network suffer from denial of service.

In Subfig. 2(b), we measure the energy consumption against number of malicious nodes and attack rate ($r_{atk}$) under energy depletion attack (EDA). In this paper, attack rate $r_{atk}$ indicates how frequently a malicious node generates and sends an attack packet to a destination node. As shown in Subfig. 2(b), RPL without adversary provides the lowest energy consumption. Since the legitimate packet sender periodically generates and sends the packets to destination nodes, each node located along the forwarding path receives and forwards a limited number of packets, the lowest energy consumption is observed. As the number of malicious nodes increases, the energy consumption of EDA increases linearly. This is because more attack packets are generated and forwarded by larger number of malicious nodes, a significant amount of energy is consumed due to receiving and forwarding a large number of packets. As the attack rate $r_{atk}$ increases, the energy consumption is significantly increased. The malicious node can generate and send attack packets more frequently with larger $r_{atk}$, thus, each intermediate node located along the forwarding path has to receive and forward a larger number of packets, which consumes more energy.

## IV. The Proposed Misbehavior-Aware Detection Scheme

In this section, we first present both system and adversary models, and then propose a misbehavior-aware detection scheme, called *MAD*, against energy depletion attack.

### A. System and Adversary Models

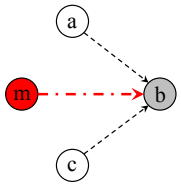
In this paper, a low power and lossy network running with RPL is considered, where a set of resource-constrained nodes communicates directly or indirectly through lossy links. During a network deployment phase, each node receives a unique identifier, e.g., an IPv6 address. Each node is also aware of its one-hop neighbor nodes by exchanging a one-time single-hop *Hello* packet piggybacked with its node id [37]. To deliver a packet toward a destination node, the packet sender employs the point-to-point (P2P) routing mechanism in RPL.

Due to the lack of physical protection, an adversary is able to capture and compromise a legitimate node, gain access to all stored information including public and private keys, and reprogram it to behave maliciously [9]. However, we do not consider node capture attack [38], where an adversary can capture a legitimate node from the network as the first step for further different types of attacks. The primary goal of the adversary is to deplete the scarce energy resource of legitimate nodes by exploiting the vulnerability of network routing protocol, and finally make the network suffer from denial of service. The malicious node may inject bogus messages into the network to consume the scarce network resource (i.e., bogus data injection attack), however, this attack can be easily prevented by using the technique proposed in [13]. In this paper, we primarily focus on the forwarding misbehavior or the adversarial scenario where a malicious node intentionally generates and sends a large number of packets to legitimate node to excessively consume the energy resource of intermediate nodes located along the forwarding path. We also assume that the network is free of other general attacks such as sybil attack, collision or jamming attack, or wormhole attack.

### B. MAD: Misbehavior-Aware Detection

The basic idea of *MAD* is that each node maintains a count of the number of received packets from its one-hop neighbor node within a specific time window, and then compares the count with a dynamically calculated threshold to detect potential malicious nodes. In this section, we investigate three major issues to implement the *MAD* scheme: (i) what information should be maintained in each node; (ii) how to detect potential malicious nodes launching energy depletion attack; and (iii) how to isolate the suspected malicious nodes from the network.

First, each node maintains an Observation Table (*OT*) to record the number of received packets from each neighbor node during an observation window ($\omega$). In this paper, observation window $\omega$ is designed as a system parameter and can be configured depending on the urgency of removing malicious nodes from the network. For example, a communication critical network in industrial control system or battlefield, $\omega$ is given a smaller value in order to frequently evaluate the forwarding operations of neighbor nodes, and quickly detect and isolate the potential adversary from the network. To balance the trade-off between detection accuracy and isolation latency, $\omega$ can have a relatively large value in non-critical situation [39]. Here, the performance impacts of $\omega$ are observed in Section V. The Observation Table *OT* consists of three components: neighbor node's id ($nid$), the number of received packets within observation window ($rp$), and the beginning timestamp of observation window ($ts$). At the beginning of each observation window, the number of received packets $rp$ is reset to zero, and the timestamp $ts$ is set to current time ($t_{cur}$). For example, in Fig. 3, when $n_m$ generates and sends a packet to $n_b$, $n_b$ records this forwarding operation and increases the number of received packets from $n_m$ by one, $OT_b[m].rp$ + 1.




Observation Table (OT)

| *nid* | *rp* | *ts* |
|---|---|---|
| a | 5 | $t_{cur}$ |
| m | 15 | $t_{cur}$ |
| c | 4 | $t_{cur}$ |

Fig. 3. A snapshot of network, where a malicious node $n_m$ generates and sends a large number of packets to next-hop node $n_b$. The Observation Table of node $n_b$ stores the forwarding record of its neighbor nodes.

Second, we also suggest each node to maintain a Detection Table (*DT*) to record the number of detected forwarding misbehaviors of each neighbor node, an entry of *DT* consists of two components: neighbor node's id ($nid$) and the number of detected forwarding misbehaviors ($c_{mis}$). If the number of received packets $rp$ from neighbor node within $\omega$ is larger than the dynamically calculated threshold, the corresponding forwarding operations of neighbor node within $\omega$ is suspected as forwarding misbehavior, and $c_{mis}$ is increased by one. In this paper, the number of detected forwarding misbehaviors $c_{mis}$ is utilized to calculate the threshold value, and indicates how much weight a neighbor node's forwarding operations $rp$ accounts for the calculation of threshold value. If a malicious node performs more forwarding misbehaviors and receives a larger $c_{mis}$, the number of received packets from the malicious node within observation window will have less weight in the calculation of threshold value, and vice versa. Note that the rationale behind this design is to consider an implicit penalty of forwarding misbehaviors. In particular, if a malicious node shows more forwarding misbehaviors which can be detected, a larger $c_{mis}$ will be observed. However, the larger $c_{mis}$ makes the $rp$ of malicious node have less weight in the calculation of threshold, thus, the threshold will be scaled to the $rp$ of normal neighbor nodes, and the forwarding misbehavior can be easily detected.

Third, at the end of each observation window, each node examines Observation Table *OT* and Detection Table *DT*, and calculates a threshold value as the reasonable number of received packets from neighbor node within observation window. In this paper, the threshold value ($T_{pkt}$) is calculated based on the historical detection result and most recent forwarding record, and it is expressed as

$$T_{pkt} = \frac{\sum_{i=nid}^{G} wt_i \cdot rp_i}{|G|}. \quad (1)$$

Here, $G$ is the one-hop neighbor list. $wt_i$ is the weight that the forwarding record of node $n_i$ accounts for the calculation of $T_{pkt}$, and it is expressed as,

$$wt_i = 1 - \frac{c^i_{mis}}{\sum_{j=nid}^{G} c^j_{mis}}. \quad (2)$$

Thus, the threshold value $T_{pkt}$ can be expressed as

$$T_{pkt} = \frac{\sum_{i=nid}^{G} \left(1 - \frac{c^i_{mis}}{\sum_{j=nid}^{G} c^j_{mis}}\right) \cdot rp_i}{|G|}. \quad (3)$$

**Notations:**

- $\omega$, $\varphi$, *OT*, *rp*, *nid*, *DT*, $wt_j$, $T_{pkt}$, and $c_{mis}$: Defined before.
- $G_i$, $D_i$, and $n_i^{pre}$: The one-hop neighbor node list of node $n_i$, the descendant node list of node $n_i$, and the preferred parent node of node $n_i$.
- $M_s$ and $M_{ns}$: The storing and non-storing mode of RPL.
- $pkt[src, des, seq, type]$: A packet containing a source node id (*src*), a destination node id (*des*), sequence number (*seq*), and packet type (*type*). Here, *type* can be *data* or *Isolate*. If the *type* is *Isolate*, the *des* field is the identifier of suspected malicious node.

**Event-driven Algorithm:**

◇ When a node $n_s$ has a packet, $pkt[s, d, seq, data]$, to destination node $n_d$:
  **if** $n_d \in G_s$
    Forward $pkt$ to $n_d$;
  **else**
    Forward $pkt$ to $n_s^{pre}$;
◇ When a node $n_j$ receives a packet, $pkt[src, des, seq, data]$, from node $n_m$:
  $OT_j[m].rp$ += 1;
  **if** $M_s$ is **true**
    **if** $des \in D_j$
      Forward $pkt$ to $n_{des}$ through cached route;
    **else**
      Forward $pkt$ to $n_j^{pre}$;
  **if** $M_{ns}$ is **true**
    **if** $des \in G_j$
      Forward $pkt$ to $n_{des}$;
    **else**
      Forward $pkt$ to $n_j^{pre}$;
◇ When an observation window ends at legitimate node $n_i$:
  **for** $n_j \in DT$
    $wt_j$ = 1 - $\frac{c^j_{mis}}{\sum_{k=nid}^{G_i} c^k_{mis}}$; /* Eq. 2 */
  $T_{pkt}$ = $\frac{\sum_{k=nid}^{G_i} wt_k \cdot rp_k}{|G|}$; /* Eq. 3 */
  **for** $n_j \in OT$
    **if** $OT_i[j].rp > T_{pkt}$
      $c^j_{mis}$ += 1;
      **if** $c^j_{mis} >= \varphi$
        Broadcast $pkt[i, j, seq, Isolate]$;

Fig. 4. The pseudo code of the proposed *MAD* scheme.

Fourth, when the observation window ends, if the number of received packets from neighbor node within observation window is larger than $T_{pkt}$, the corresponding forwarding operations are suspected as forwarding misbehavior, and the number of detected forwarding misbehaviors, $c_{mis}$, is increased by one. In addition, when the number of detected forwarding misbehaviors of the suspected node reaches a threshold ($\varphi$), the detecting node broadcasts an *Isolate* packet to its one-hop neighbor nodes to prevent them from receiving or accepting any packet from the suspected malicious node. This way, the malicious node cannot be involved in any routing operations, and it is isolated and removed from the network. For example in Fig. 3, suppose that $c^a_{mis}$ = 1, $c^m_{mis}$ = 3, and $c^c_{mis}$ = 1. Here, $c_{mis}$ of each node is initially set to 1. According to Eq. 2, $wt_a$ = 0.8, $wt_m$ = 0.4, and $wt_c$ = 0.8, respectively. Thus, based on Eq. 3, $T_{pkt}$ = $\frac{(0.8\times5)+(0.4\times15)+(0.8\times4)}{3}$ = 4.4. As a result, the forwarding operations of malicious node $n_m$ within current $\omega$ is suspected as forwarding misbehavior by $n_b$, and $c^m_{mis}$ is increased by one. This is because the total number of generated and forwarded packets by $n_m$ is 15, which is larger than the calculated threshold $T_{pkt}$ = 4.4. The malicious node could behave like normal nodes, and generates and sends a small number of packets, however, it does not benefit from

doing so because a small number of packets will not cause a significant amount of energy consumption. Major operations of the *MAD* scheme are summarized in Fig. 4.

## V. Performance Evaluation

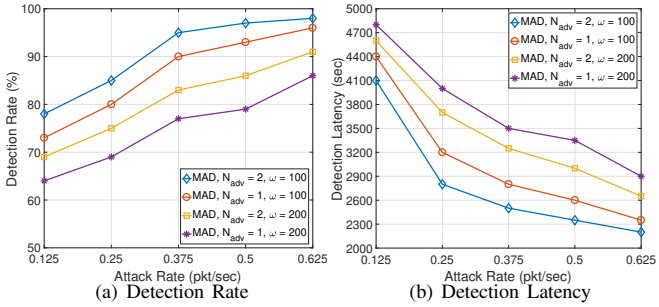
### A. Simulation Testbed

We conduct extensive simulation experiments using OMNeT++ [14] to evaluate the performance of the proposed scheme. 100 nodes including one DODAG root are uniformly distributed within a 200×200 $m^2$ square network area. The communication range of each node is 30 (m). An exponential packet rate with mean 0.1 is adopted to emulate low network traffic scenarios in LLNs. The size of each packet is 40 bytes. The radio model simulates CC2420 with a normal data rate of 250 Kbps, and 802.15.4 MAC/PHY operates with a default configuration in the 2.4 GHz band [40]. The channel error rate is set to 10%. We assume that the DODAG root is always trusted, and a couple of legitimate nodes are compromised and reprogrammed by adversary to behave maliciously. The attack rate varies between 0.125 and 0.625 pkt/sec. The total simulation time is 5000 seconds. In this paper, we measure the performance in terms of detection rate, detection latency, node behavior distribution, energy consumption, and packet delivery ratio by changing key simulation parameters, including number of malicious nodes ($N_{adv}$), attack rate ($r_{atk}$), and observation window ($\omega$). For performance comparison, we compare the proposed *MAD* scheme with standard RPL routing protocol with and without adversary, respectively.

### B. Simulation Results and Analysis

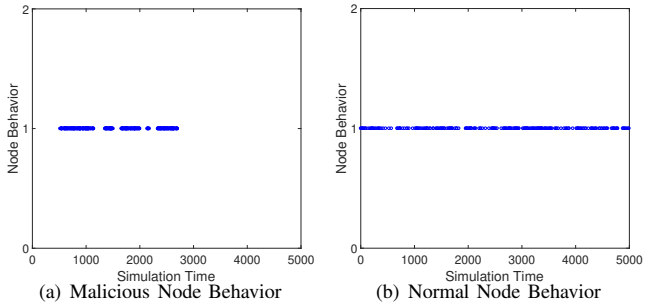
In Subfig. 5(a), we measure the detection rate by changing attack rate $r_{atk}$, number of malicious nodes $N_{adv}$, and observation window $\omega$. As the attack rate increases linearly, the detection rate of the proposed scheme increases quickly. This is because the malicious node generates and sends a larger number of attack packets due to larger $r_{atk}$, which is much higher than normal packet rate, the packet receiver can easily detect the forwarding misbehavior by comparing forwarding record with the calculated threshold with a significant difference. When the attack rate reaches 0.625 pkt/sec, the overall detection rate is above 85%. When the number of malicious nodes $N_{adv}$ increases to 2, a higher detection rate is observed. This is because more malicious nodes show forwarding misbehaviors by generating and sending attack packets, more forwarding misbehaviors can be detected, and finally a higher detection rate can be observed. When a smaller observation window $\omega$ is configured in the approach, a higher detection rate can be achieved. Since the forwarding record of neighbor nodes can be evaluated more frequently with smaller $\omega$, more forwarding misbehaviors can be detected.

In Subfig. 5(b), we measure the detection latency by changing attack rate $r_{atk}$, number of malicious nodes $N_{adv}$, and observation window $\omega$. As the attack rate increases, the detection latency significantly decreases. This is because the malicious node generates and sends more attack packets with larger attack rate, more forwarding misbehaviors can be easily




(a) Detection Rate (b) Detection Latency

Fig. 5. The detection rate and detection latency against attack rate.



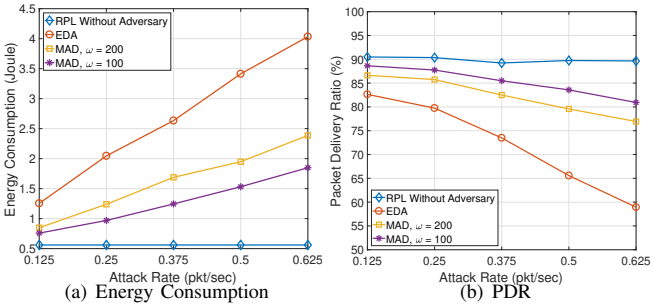

(a) Malicious Node Behavior (b) Normal Node Behavior

Fig. 6. The node behaviors against simulation time.

detected, and the malicious node can be removed from the network more quickly. With a shorter observation window $\omega$, a lower detection latency is observed. This is because the malicious node will be evaluated more often with smaller $\omega$, more forwarding misbehaviors can be detected, and a lower detection latency is achieved. When more malicious nodes exist in the network, a lower detection latency can be achieved.

In Fig. 6, the malicious and legitimate source node's behaviors are observed against simulation time. Here, 1 indicates the behavior of generating a packet. In Fig. 6(a), the packet generating behaviors are intermittent and mainly concentrate in several small time periods. Since the malicious node generates and sends a large amount of packets within a short time period, this forwarding misbehavior can be easily detected by the proposed scheme. In addition, as shown in Fig. 6(a), a series of packet generating behaviors are observed from 500 sec to approximately 2700 sec. Then no packet generating behaviors are observed, which indicates that the *MAD* successfully isolates the malicious node from the network around 2700 sec. In Fig. 6(b), the packet generating behaviors of normal node are evenly distributed in the entire simulation period.

In Subfig. 7(a), the energy consumption is measured based on the number of forwarded and overheard packets [41] by changing attack rate $r_{atk}$ and $\omega$. Without adversary, the energy consumption of RPL is maintained around 0.5 Joule. This is because a low packet rate is employed by legitimate nodes, which generate and send a limited number of packets to destination nodes, the lowest energy consumption is observed. As the attack rate increases, the energy consumption of energy depletion attack (EDA) significantly increases. This is because the malicious node generates and sends more attack packets to destination nodes, all the intermediate nodes located along the




(a) Energy Consumption (b) PDR

Fig. 7. The energy consumption and packet delivery ratio against attack rate.

forwarding path have to receive and forward a large number of packets, which consume a large amount of energy resource. However, as shown in Subfig. 7(a), the *MAD* can significantly reduce the energy consumption. Since each node counts the number of received packets from its one-hop neighbor nodes, calculates the threshold of the number of received packets within an observation window, and detects the forwarding misbehaviors of malicious node. Thus, the malicious node can be isolated from the network quickly when the number of detected forwarding misbehaviors reaches a threshold value, and the network traffic is significantly reduced.

In Subfig. 7(b), we measure the packet delivery ratio (PDR) by changing attack rate $r_{atk}$ and $\omega$. The RPL without adversary achieves the highest PDR (about 90%), this is because the legitimate node generates and sends a limited number of packets, and every node cooperatively and faithfully forwards the received packets to destination node. However, due to bad channel quality, a few number of packets (approximate 10%) could get lost. As the attack rate increases, the PDR of EDA significantly decreases. Since each intermediate node located along the forwarding path receives and forwards a large number of packets, which depletes all energy resource, the legitimate nodes cannot receive any packet further. The *MAD* achieves a higher PDR than that of EDA. This is because the malicious node can be isolated from the network quickly, the legitimate nodes will be able to involve in the packet forwarding and receiving operations. As the observation window extends, a higher PDR can be observed.

## VI. Discussion

In this section, we first investigate the applicability of *MAD* to other attacks and then further explore its design issues and extensions for future research.

### A. Immunity to Other Attacks

We investigate the *MAD* to see whether it can be applied to other two well-known attacks: DIO suppression attack [15] and DAO inconsistency attack [42].

*1) DIO Suppression Attack:* A malicious node can periodically replay previously overheard DIO messages to induce victim nodes to suppress the transmission of DIO messages, which are the RPL control messages necessary to build the routing topology. The DIO suppression attack can cause a degradation of the routes' quality, and eventually leads to network partitions. Unlike other RPL attacks, the DIO suppression attack does not require the adversary to steal cryptographic keys from legitimate nodes. Thus, it is not trivial to avoid DIO suppression attack, but this attack can be detected by the *MAD*. For example in Fig. 3, suppose $n_m$ periodically broadcasts the overheard DIO messages with a fixed interval to suppress the transmission of DIO message of $n_b$. In the *MAD*, $n_b$ can maintain a count of the number of received DIO messages from one-hop neighbor nodes (e.g., $n_a$, $n_c$, and $n_m$) within an observation window, and then calculates a threshold value. If the number of DIO messages issued by $n_m$ is larger than the threshold value, $n_m$ is suspected as malicious node.

*2) DAO Inconsistency Attack:* In RPL, each node in storing mode can quickly learn the routes of its descendants by aggressively caching the piggybacked downward route information of received DAO messages in its routing table. If a parent node receives a forwarding error packet from its child node corresponding to previously forwarded data packet, the parent node removes the cached downward route designated to the destination node via this child node from routing table. Thus, a malicious node can intentionally drop the received data packets and replies a large number of forwarding error packets to cause the parent node to empty the downward routing table. However, this attack can be detected by the *MAD*. In Fig. 3, suppose $n_m$ drops all the received data packets and replies a large number of forwarding error packets to the parent node $n_b$. However, $n_b$ can record all the received forwarding error packets from all child nodes within an observation window, and then computes a threshold of forwarding error rate of its neighbor nodes. If $n_m$ shows larger forwarding error rate than the threshold value, it will be suspected as malicious node. Thus, the DAO inconsistency attack can be easily detected.

### B. Potential Enhancements

In the *MAD*, each node maintains the number of received packets from its one-hop neighbor node within a specific time window, and compares the count with a dynamically calculated threshold to detect the potential malicious node. When the number of detected misbehaviors of suspected node reaches a threshold, the node broadcasts an *Isolate* packet to its one-hop neighbor nodes to prevent them from receiving or accepting any packet from the suspected malicious node. In order to timely reduce the impacts of energy depletion attack, we plan to design a packet rate control technique based on [43] by specifying the maximum number of accepted packets for forwarding in each observation window. For example in Fig. 3, at the end of observation window, $n_b$ calculates a threshold as the reasonable number of received packets from neighbor nodes, and uses this threshold as the maximum number of accepted packets for forwarding from each neighbor node in the next observation window. Thus, the number of forwarded packets originated from suspected malicious node can be significantly reduced, each intermediate node located along the forwarding path will receive and forward a less number of packets, resulting in less energy consumption. Finally, the impact of energy depletion attack can be timely reduced.

## VII. Conclusion

In this paper, we first present and analyze the energy depletion attack with a preliminary result in RPL-based LLNs. Then we propose a misbehavior-aware detection scheme to efficiently detect the energy depletion attack, where each node maintains a count of the number of received packets from its neighbor node within a specific time window, and then compares the count with a dynamically calculated threshold to detect the potential malicious node. Extensive simulation results indicate that the proposed scheme is a viable approach against energy depletion attack in RPL-based LLNs. Since radio propagation and its channel dynamics cannot easily be captured by simulation models, to see the full potential of the proposed countermeasure, we plan to develop a small-scale testbed and deploy a real network composed of TelosB motes in an indoor office environment for experimental study.

## Acknowledgment

This research was supported by NASA West Virginia Space Grant Consortium, Training Grant #NNX15AI01H, and Undergraduate Creative Discovery Scholar and Undergraduate Research Scholar Awards at Marshall University.

## References

[1] M. Palattella *et al.*, "Internet of Things in the 5G Era: Enablers, Architecture, and Business Models," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 3, pp. 510–527, 2016.

[2] *IHS. Internet of Things (IoT) Connected Devices Installed Base Worldwide*, https://www.statista.com/statistics/485136/global-internet-of-things-market-size/.

[3] S. Dawson-Haggerty, A. Tavakoli, and D. Culler, "Hydro: A Hybrid Routing Protocol for Low-Power and Lossy Networks," in *Proc. IEEE SmartGridComm*, 2010, pp. 268–273.

[4] T. Winter and P. Thubert, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks," *RFC Standard 6550*, March 2012.

[5] A. Nia and N. Jha, "A Comprehensive Study of Security of Internet-of-Things," *IEEE Trans. Emerg. Topics Comput.*, September 2017, https://10.1109/TETC.2016.2606384.

[6] "802.15.4-2011 - IEEE Standard for Local and metropolitan area networks–Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs)," September 2011.

[7] C. W. Paper, "Cisco Connected Grid Security for Field Area Network)," January 2012.

[8] H. Kim, J. Ko, D. Culler, and J. Paek, "Challenging the IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL): A Survey," *IEEE Commun. Surveys Tuts.*, September 2017, https://10.1109/COMST.2017.2751617.

[9] S. Challa, M. Wazid, A. Das, N. Kumar, A. Reddy, E. Yoon, and K. Yoo, "Secure Signature-Based Authenticated Key Establishment Scheme for Future IoT Applications," *IEEE Access*, vol. 5, pp. 3028–3043, 2017.

[10] A. Sehgal, V. Perelman, S. Kuryla, and J. Schonwalder, "Management of Resource Constrained Devices in the Internet of Things," *IEEE Commun. Mag.*, vol. 50, no. 12, p. 144–149, 2012.

[11] A. Sehgal, A. Mayzaud, R. Badonnel, I. Chrisment, and J. Schnwlder, "Addressing DODAG Inconsistency Attacks in RPL Networks," in *Proc. IEEE GIIS*, 2014, pp. 1–8.

[12] T. Tsao, R. Alexander, M. Dohler, V. Daza, A. Lozano, and M. Richardson, "A Security Threat Analysis for the Routing Protocol for Low-Power and Lossy Networks (RPLs)," *RFC Standard 7416*, January 2015.

[13] A. Abdallah and X. Shen, "Efficient Prevention Technique for False Data Injection Attack in Smart Grid," in *Proc. IEEE ICC*, 2016, pp. 1–6.

[14] A. Varga, *OMNeT++*, 2014, http://www.omnetpp.org/.

[15] P. Perazzo, C. Vallati, G. Anastasi, and G. Dini, "DIO Suppression Attack Against Routing in the Internet of Things," *IEEE Commun. Lett.*, vol. 21, no. 11, pp. 2524 – 2527, 2017.

[16] J. Heo, J. Kim, S. Bahk, and J. Paek, "Dodge-Jam: Anti-Jamming Technique for Low-Power and Lossy Wireless Networks," in *Proc. IEEE SECON*, 2017, pp. 1–9.

[17] C. Pu and S. Hajjar, "Mitigating Forwarding Misbehaviors in RPL-based Low Power and Lossy Networks," in *Proc. IEEE CCNC*, 2018, pp. 1–6.

[18] C. Pu, "Mitigating DAO Inconsistency Attack in RPL-based Low Power and Lossy Networks," in *Proc. IEEE CCWC*, 2018, pp. 570–574.

[19] C. Pu and T. Song, "Hatchetman Attack: A Denial of Service Attack Against Routing in Low Power and Lossy Networks," in *Proc. IEEE CSCloud*, 2018, pp. 12–17.

[20] C. Pu, "Energy Depletion Attack Against Routing Protocol in the Internet of Things," in *Proc. IEEE CCNC*, 2019, pp. 1–4.

[21] C. Pu, X. Zhou, and S. Lim, "Mitigating Suppression Attack in Multicast Protocol for Low Power and Lossy Networks," in *Proc. IEEE LCN*, 2018, pp. 251–254.

[22] C. Pu and X. Zhou, "Suppression Attack Against Multicast Protocol in Low Power and Lossy Networks: Analysis and Defenses," *Sensors*, vol. 18, no. 10, p. 3236, 2018.

[23] C. Pu, "Spam DIS Attack Against Routing Protocol in the Internet of Things," in *Proc. IEEE ICNC*, 2019, pp. 73–77.

[24] S. Raza, L. Wallgren, and T. Voigt, "SVELTE: Real-time intrusion detection in the Internet of Things," *Ad Hoc Networks*, vol. 11, no. 8, pp. 2661–2674, 2013.

[25] R. Hummen, J. Hiller, H. Wirtz, M. Henze, H. Shafagh, and K. Wehrle, "6LoWPAN Fragmentation Attacks and Mitigation Mechanisms," in *Proc. ACM WiSec*, 2013, pp. 55–66.

[26] S. Sajjad *et al.*, "Security analysis of IEEE 802.15. 4 MAC in the context of Internet of Things (IoT)," in *Proc. IEEE CIACS*, 2014, pp. 9–14.

[27] P. Kasinathan, C. Pastrone, M. A. Spirito, and M. Vinkovits, "Denial-of-Service detection in 6LoWPAN based Internet of Things," in *Proc. IEEE WiMob*, 2013, pp. 600–607.

[28] A. Rghioui, A. Khannous, and M. Bouhorma, "Denial-of-Service attacks on 6LoWPAN-RPL networks: Threats and an intrusion detection system proposition," *JACST*, vol. 3, no. 2, pp. 143–152, 2014.

[29] C. Pu and S. Lim, "Spy vs. Spy: Camouflage-based Active Detection in Energy Harvesting Motivated Networks," in *Proc. IEEE MILCOM*, 2015, pp. 903–908.

[30] C. Pu, S. Lim, B. Jung, and J. Chae, "EYES: Mitigating Forwarding Misbehavior in Energy Harvesting Motivated Networks," *Elsevier Computer Communications*, vol. 124, pp. 17–30, 2018.

[31] C. Pu, S. Lim, J. Byungkwan, and M. Manki, "Mitigating Stealthy Collision Attack in Energy Harvesting Motivated Networks," in *Proc. IEEE MILCOM*, 2017, pp. 575–580.

[32] C. Pu and S. Lim, "A Light-Weight Countermeasure to Forwarding Misbehavior in Wireless Sensor Networks: Design, Analysis, and Evaluation," *IEEE Systems Journal*, vol. 12, pp. 834–842, 2016.

[33] C. Pu, S. Lim, C. Jinseok, and J. Byungkwan, "Active Detection in Mitigating Routing Misbehavior for MANETs," *Wireless Network*, vol. 25, no. 4, pp. 1669—-1683, 2017.

[34] C. Pu, "Jamming-Resilient Multipath Routing Protocol for Flying Ad Hoc Networks," *IEEE Access*, vol. 6, pp. 68 472–68 486, 2018.

[35] Y. Zou, J. Zhu, X. Wang, and L. Hanzo, "A Survey on Wireless Security: Technical Challenges, Recent Advances, and Future Trends," *Proceedings of the IEEE*, vol. 104, no. 9, pp. 1727–1765, 2016.

[36] O. Gaddour and A. Koubâa, "RPL in a nutshell: A survey," *Computer Networks*, vol. 56, no. 14, pp. 3163–3178, 2012.

[37] C. Pu, T. Gade, S. Lim, M. Min, and W. Wang, "Lightweight Forwarding Protocols in Energy Harvesting Wireless Sensor Networks," in *Proc. IEEE MILCOM*, 2014, pp. 1053–1059.

[38] J. Zhao, "On Resilience and Connectivity of Secure Wireless Sensor Networks Under Node Capture Attacks," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 3, pp. 557–571, 2017.

[39] X. Li, R. Lu, X. Liang, and X. Shen, "Side Channel Monitoring: Packet Drop Attack Detection in Wireless Ad Hoc Networks," in *Proc. IEEE ICC*, 2011, pp. 1–5.

[40] A. Boulis, *Castalia*, 2014, http://castalia.forge.nicta.com.au.

[41] X. Tang and J. Xu, "Extending Network Lifetime for Precision-Constrained Data Aggregation in Wireless Sensor Networks," in *Proc. IEEE INFOCOM*, 2006, pp. 1–12.

[42] J. Hui, "The Routing Protocol for Low-Power and Lossy Networks (RPL) Option for Carrying RPL Information in Data-Plane Datagrams," *RFC Standard 6553*, March 2012.

[43] M. Handley, S. Floyd, J. Padhye, and J. Widmer, "TCP Friendly Rate Control (TFRC): Protocol Specification," *RFC Standard 5348*, January 2003.