Mitigating Forwarding Misbehaviors in RPL-based Low Power and Lossy Networks

Cong Pu
Division of Computer Science
Marshall University
Huntington, WV 25755
Email: puc@marshall.edu

Salam Hajjar
Division of Engineering
Marshall University
Huntington, WV 25755
Email: hajjar@marshall.edu

***Abstract*—Low power and lossy networks (LLNs) are rapidly emerging as an important part of ubiquitous computing, and serving as a major building block for the communication infrastructure in the presence of Internet-of-Things (IoT). Routing protocol for low power and lossy networks (RPL) is a novel routing protocol standardized to enable the integration of resources-constrained devices into the Internet. However, due to the shared radio medium, the lack of physical protection and security requirements of inherent routing protocol, low power and lossy networks are admittedly threatened by diverse Denial-of-Service (DoS) attacks that primarily disrupt network protocols and interfere with on-going communications. In this paper, we propose a monitor-based approach, called *CMD*, to mitigate forwarding misbehaviors in LLNs running with RPL, where single or multiple malicious nodes randomly or strategically drop any incoming *Data* packet. The basic idea of the CMD is that each node monitors the forwarding behaviors of the preferred parent node to observe the packet loss rate, compares the observation result with the collected packet loss rate from one-hop neighbor nodes, and detects the forwarding misbehaviors of the preferred parent node. We evaluate the proposed scheme through extensive simulation experiments using OMNeT++ and compare its performance with the original RPL protocol and the existing two-step detection scheme. The simulation results show that the proposed scheme can not only improve the detection rate and packet delivery ratio (PDR) but also can reduce the energy consumption and isolation latency.**

***Index Terms*—Forwarding misbehaviors, monitor-based detection, low power and lossy networks.**

I. Introduction

Internet-of-Things (IoT) and its applications are rapidly proliferating, where a myriad of multi-scale sensors and devices (later, nodes) are seamlessly blended [1]. As a part of speedily emerging IoT, it is envisaged that low power and lossy networks (LLNs), where a set of resources-constrained nodes with limited processing power, memory storage and battery energy communicates directly or indirectly via lossy links, will play an important role in building a ubiquitous computing and communication infrastructure. With the increasing demand for resources-constrained nodes to be connected to Internet, routing protocol for low power and lossy networks, referred to as RPL [2], has been standardized and receiving a considerable attention as the communication standard for smart node networks. With the prevalence of data mining techniques, cloud computing and social networking paradigms as well as the remarkable recent progress in computing power, sensors and embedded devices, and wireless communications and networking technologies, we envision that wirelessly connected smart nodes under IoT will enhance flexible information accessibility and availability as well as change our lives further.

Due to the lack of physical protection and tamper resistance, nodes in LLNs can be easily captured, tampered, or destroyed by an adversary. Although RPL provides optional cryptography mechanisms to verify the authenticity and integrity of control messages while providing confidentiality [2], a legitimate node compromised by an adversary can still overhear, duplicate, corrupt, or alter an on-flying packet because of open nature of wireless communication. In addition, the RPL security services proposed in [2], [3] do not address all possible attacks and are subjected to some threats that may compromise RPL security to disrupt routing protocol or interfere with on-going communications [4]. For example, in RPL, a malicious node can intentionally advertise a better rank thus making nodes in the DODAG select it as the preferred parent node, and then selectively or randomly drop any incoming *Data* packet on purpose to deafen an intended DODAG root (or sink).

In this paper, we investigate the forwarding misbehavior and propose its countermeasure in LLNs running with RPL, where single or multiple malicious nodes forward all control packets but randomly or strategically drop any incoming *Data* packet. This kind of forwarding misbehavior primarily targets the network routing vulnerabilities of multi-hop wireless networks by violating an implicit assumption, i.e., all nodes faithfully and collaboratively route *Data* packets to a destination. Unlike a blackhole attack [3], where a malicious node blindly drops any incoming packet, it is not trivial to detect this forwarding misbehavior from temporal node failures or normal packet losses. In light of this, we propose a monitor-based detection approach and its corresponding techniques to efficiently mitigate the forwarding misbehaviors. Our major contribution is briefly summarized in twofold.

- First, we propose a monitor-based detection approach, called *CMD*, in LLNs running with RPL. The basic idea is that each node monitors the forwarding behaviors of the preferred parent node to observe the packet loss rate, and then compares the observation result with the collected packet loss rate from one-hop neighbor nodes to detect

978-1-5386-4790-5/18/$31.00 ©2018 IEEE

the forwarding misbehaviors of the preferred parent node. The CMD is also incorporated with timeout and one-time retransmission techniques to recover unexpected packet losses due to forwarding misbehaviors or bad channel quality.
- Second, we revisit and implement a prior two-step detection approach [5] and the original RPL [2] without detection mechanism for performance comparison. In addition, the original RPL is used as the lower bound of packet delivery ratio and energy consumption.

We develop a customized discrete event-driven simulation framework by using OMNeT++ [6] and evaluate its performance through extensive simulation experiments in terms of detection rate, packet delivery ratio, energy consumption, and isolation latency. The simulation results indicate that the proposed countermeasure is a viable detection approach to forwarding misbehaviors in LLNs running with RPL.

The rest of the paper is organized as follows. Prior schemes are summarized and analyzed in Section II. The background work and our motivation with a preliminary result are presented in Section III. The proposed countermeasure and performance evaluation with extensive simulation experiments are presented in Sections IV and V, respectively. Finally, concluding remarks are provided in Section VI.

II. Related Work

In [7], a novel intrusion detection system, called SVELTE, is proposed to secure IPv6 over Low-power Wireless Personal Area Network (6LoWPAN) running with RPL from network layer and routing attacks. In the SVELTE, 6LoWPAN Mapper gathers information about the RPL network and reconstructs the network in the border router, and the intrusion detection module integrated with mini-firewall analyzes the traffic and detects intrusion.

In [8], a rank attack that aims at the rank property in the RPL and its impact on the performance are investigated in wireless sensor networks, where the attacker can compromise the rank rule to downgrade the RPL performance. Four adversarial scenarios motivated by violating rank rule permanently and non-permanently and their potential performance impact are analyzed. In the VeRA [9], a version number and rank authentication security scheme based on one-way hash chains are proposed to secure the RPL in LLN, where the misbehaving nodes illegitimately increase the version number of DIO message and compromise illegal rank values. In order to protect against the attackers that send DIO messages with higher version number values or that publish a high rank value, the version numbers are binded with authentication data and signatures. In [10], a topological inconsistency attack that degrades channel availability and increases energy consumption is investigated in RPL-based LLNs, where a malicious node manipulates the packet header options and forwards it to the next-hop node to drop the modified packet. In order to mitigate such attack, an adaptive threshold mechanism with the consideration of dynamic network characteristics is proposed.

A camouflage-based approach is proposed to detect the selective forwarding attack in energy harvesting motivated networks in [11]. Each node actively disguises itself as an energy harvesting node on purpose and pretends not to overhear, and then stealthily monitors any forwarding operation of its adjacent nodes to detect a lurking malicious node. In the CRS-A [12], each node maintains a reputation table with an adaptive detection threshold to evaluate the forwarding behavior of its adjacent nodes in wireless sensor network. The reputation value is calculated based on the deviation of the monitored packet loss rate as well as the estimated normal loss rate caused by the time- and location-variant channel quality and the link layer collisions. The node with low reputation value is detected and isolated from the routing path. [5] proposes a two-step detection approach consisting of local monitoring and global verification against blackhole attack in RPL-based networks, where a malicious node silently drops all the received packets. In [5], each node observes the communication behaviors of its neighbor nodes and counts any misbehaving activity. If the recorded misbehaving activities exceed the threshold, the monitoring node sends the verification packet to the DODAG root to verify whether the sending or forwarding packet was received or not.

In summary, forwarding misbehaviors and their diverse countermeasures have been well studied in various networks. However, to the best of our knowledge, little attention has been paid for resources-constrained devices in the realm of low power and lossy networks running with RPL.

III. Background and Motivation

In this section, we briefly review the basic operations of RPL, investigate a potential forwarding misbehavior and measure its performance impact on RPL with a preliminary result.

A. The RPL Routing Protocol

RPL [2] is a novel routing protocol designed for low power and lossy networks by the Internet Engineering Task Force (IETF) Working Group. The basic idea of RPL is to use one or more Destination-Oriented Directed Acyclic Graphs (DODAGs) to maintain the network state information. Each DODAG is a directed graph associated with one DODAG root, where all edges are oriented toward the DODAG root and no cycles exist. The DODAG root can be either a base station or gateway node that acts as a data sink, generates a new DODAG that trickles downward to leaf nodes, and bridges the LLNs with IPv6 networks. Each node has a rank that implies the node's individual position relative to other nodes with respect to a DODAG root, and the rank value is determined based on the Objective Function which describes the routing metrics and constraints. To avoid any routing loop, the rank of nodes along any path to the DODAG root should be monotonically decreasing.

In order to construct a DODAG, the DODAG root will broadcast a DAG Information Object (DIO) control message, which includes a DODAG root ID, the rank of the DODAG

root, and an Objective Function. Any other node that receives a DIO message and is willing to join the DODAG should add the DIO message sender to its parent list, compute its own rank value according to the piggybacked Objective Function, and pass on the DIO message with the updated rank information. The node that has the lowest rank value among all the nodes in the parent list is selected as the preferred parent node; any *Data* traffic destined to the DODAG root will be forwarded by choosing the preferred parent node as the next-hop forwarding node. Each node can issue a Destination Advertisement Object (DAO) control message to propagate reverse route information and record the nodes visited along the upward path to DODAG root. After passing the DAO message through the path from a particular node to the DODAG root, a complete path between the DODAG root and the node is established. If a new node wants to join the network, it can request topology information from the neighboring nodes in adjacent DODAGs by issuing a DAG Information Solicitation (DIS) control message.

B. Potential Forwarding Misbehavior

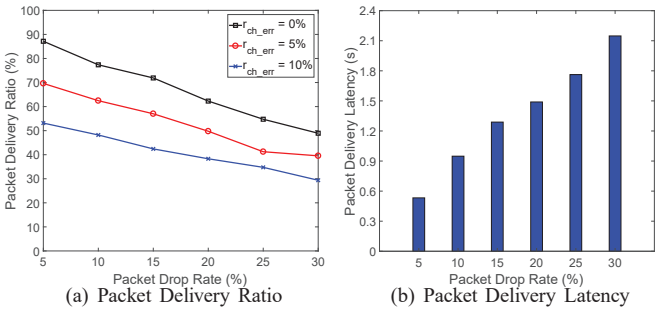
Although RPL provides optional cryptography mechanisms to support message authenticity, confidentiality and integrity and protect the network against the external attacker, the internal attacker can still compromise the RPL cryptography defense and downgrade the performance through interrupting on-going communication [13]. For example, a compromised legitimate node can forward all control packets but randomly or strategically drop any incoming *Data* packet to deafen the intended DODAG root. In Fig. 1, we measure the packet delivery ratio (PDR) and packet delivery latency by varying packet drop rate and channel error rate (r_{ch_err}). As shown in Subfig. 1(a), the PDR decreases as the packet drop rate increases. This is because more *Data* packets are dropped by malicious nodes, less number of *Data* packets can reach the DODAG root. The r_{ch_err} significantly affects the PDR. As the r_{ch_err} increases, overall PDR decreases because *Data* packets can be lost due to bad channel quality, resulting in lower PDR. According to Subfig. 1(b), the packet delivery latency increases when the packet drop rate increases. This is because a larger number of *Data* packets are dropped by malicious nodes as the packet drop rate increases, more *Data* packets experience the packet delivery timeouts, thus overall packet delivery latency increases.

IV. Proposed Countermeasure

In this section, we first present the adversarial model and then propose a monitor-based approach, called *CMD*, to mitigate the forwarding misbehaviors in low power and lossy networks running with RPL.

A. Adversary Model

We consider a low power and lossy network running with RPL, where a set of resources-constrained nodes and one DODAG root communicate directly or indirectly through lossy links. Each node is uniquely identified by a node ID, e.g., an Internet Protocol (IP) address of the node. For the simplicity,



(a) Packet Delivery Ratio (b) Packet Delivery Latency

Fig. 1. The performance of packet delivery ratio and packet delivery latency against packet drop rate and channel error rate. Here, we consider a square network (200×200 (m^2)), where 100 nodes are uniformly distributed.

we assume that the RPL only maintains one DODAG structure rooted at the DODAG root in this paper. 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. The primary goal of the adversary is to disrupt network routing protocols and interfere with on-going communication. The malicious node may eavesdrop on an on-flying packet and inject false information or modify its packet header to mislead network traffic. If a sender can authenticate a packet with a light-weight digital signature [14], a receiver can easily verify the packet and detect any modification. A malicious node will not blindly drop all incoming packets (i.e., blackhole attack) because its child nodes may consider it as a failed node and select an alternative parent node. However, the malicious node can collaboratively and faithfully forward all routing control packets but randomly or strategically drop any incoming *Data* packet. In this paper, we primarily focus on the forwarding misbehaviors or the adversarial scenarios where single or multiple malicious nodes selectively or randomly drop any incoming *Data* packet to deafen the DODAG root. We assume that the system is free of other general attacks such as sybil attack, collision or jamming attack, or wormhole attack. We do not consider cryptographic primitives.

B. CMD: Monitor-based Detection

The basic idea of *CMD* is that each node monitors the forwarding behaviors of the preferred parent node to observe the packet loss rate, and then compare the observation result with the collected packet loss rate from one-hop neighbor nodes to detect the forwarding misbehaviors of the preferred parent node.

First, each node maintains a forwarding record (*FR*) to store the historical statistics of forwarding to its preferred parent node, where the forwarding record consists of five components: preferred parent node's id (*rid*), the number of forwarded *Data* packets (*fp*), the number of unoverheard *Data* packets (*uf*), evaluation window (ω), and the beginning timestamp of the evaluation window (*ts*). Here, evaluation window ω is a system parameter and its impact on the performance is observed in Section V. For example, suppose a node n_s forwards a *Data* packet to its preferred parent node n_m as shown in Subfig. 2(a), thus n_s increases $FR_s[m].fp$

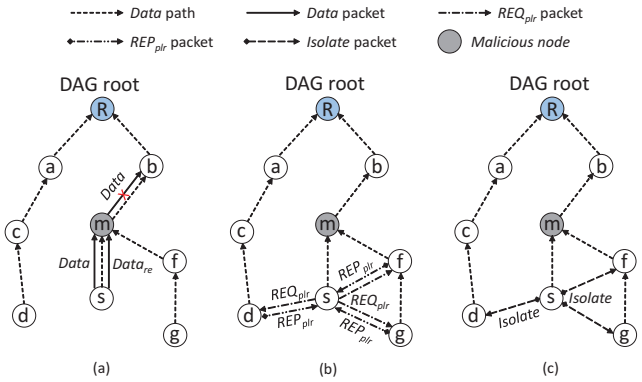


Fig. 2. A set of snapshots of the proposed CMD scheme.

by one and set *ts* to the current time (t_{cur}). However, if the malicious node n_m drops the received *Data* packet without forwarding, n_s cannot overhear the packet forwarded before the timeout period and then increases $FR_s[m].uf$ by one.

Second, when a node forwards a *Data* packet, it sets up a timer for overhearing the subsequent *Data* packet forwarded by the preferred parent node. If the node does not overhear the subsequent *Data* packet forwarded before its timer expires, because of malicious packet drop or bad channel quality, it increases the number of unoverheard *Data* packet *uf* by one. In the CMD, we propose a simple timeout technique to detect possible packet loss due to malicious packet drop or bad channel quality. We define T^O as the timeout period of overhearing *Data* packet forwarding from the preferred parent node. In order to estimate the timeout period, we consider a single-hop average trip time of overhearing *Data* packet forwarding (T^O_{avg}), which can be measured by the time from when a node forwards a *Data* packet ($T_{F,Data}$) to when it overhears the *Data* packet forwarded ($T_{O,Data}$). T^O_{avg} is updated by the low-pass filter with a filer gain constant α.

$$T^O_{avg} = \alpha \cdot T^O_{avg} + (1-\alpha) \cdot T^O_{k-1} \quad (1)$$

$$T^O_{avg} = \frac{\sum_{i=1}^{k-2} T^O_i}{k-2} \quad (2)$$

T^O_{k-1} is the measurement from the most recently overheard *Data* packet forwarding, which is expressed as

$$T^O_{k-1} = T_{O,Data} - T_{F,Data} \quad (3)$$

Thus, the timeout period is expressed as

$$T^O = T^O_{avg} + T^O_{avg} \cdot \delta, \quad (4)$$

where δ is an adjustment factor and $T^O_{avg} \cdot \delta$ is added to consider the packet forwarding latency. Fig. 3 shows the observed single-hop average trip time (or timeout period) against the simulation time.

Third, we deploy a one-time retransmission technique to remedy the *Data* packet loss and expedite in detecting the malicious nodes in the CMD. If the node does not overhear the *Data* packet forwarded before its timer expires, it increases uf by one and retransmits the *Data* packet to the preferred parent node. However, it does not increase fp since the retransmitted *Data* packet is not new one. If the node still cannot overhear

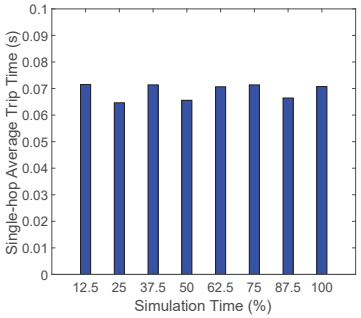


Fig. 3. The single-hop average trip time against simulation time.

the *Data* packet forwarded by the preferred parent node, it increases uf by one again and quits the retransmission. For example, as shown in Subfig. 2(a), suppose n_m drops the *Data* packet from n_s, thus n_s cannot overhear the *Data* packet forwarding. When the timeout period expires, n_s increases $FR_s[m].uf$ by one and retransmits the *Data* packet, $Data_{re}$, to n_m. If n_m drops the *Data* packet again, n_s increases $FR_s[m].uf$ by one again and quits the retransmission. The essence of one-time retransmission technique is that the more malicious node drops the retransmitted *Data* packet, the sooner the malicious node can be detected.

Fourth, when an evaluation window ω ends, each node computes the packet loss rate of the preferred parent node, $r_{los} = \frac{uf}{fp}$, based on the forwarding record *FR*. If the r_{los} is larger than or equal to a threshold (T_{ch_loss}), e.g., 10% channel error rate according to the results shown in [15], the node broadcasts a packet loss rate request (REQ_{plr}) packet to all its one-hop neighbor nodes. If the r_{los} is smaller than T_{ch_loss}, the node resets the *ts* to the current time t_{cur}. As shown in Subfig. 2(b), after broadcasting a REQ_{plr} packet by n_s, any one-hop neighbor node (i.e., n_d, n_f, or n_g) that receives the REQ_{plr} packet replies the observed packet loss rate of its preferred parent node (REP_{plr}) packet to n_s after a random backoff period. After a period of timeout T^{REP}, the node computes the local packet loss rate (r_{n_los}), which is the average value of all the received packet loss rates from its one-hop neighbor nodes. If the r_{los} is larger than r_{n_los}, the preferred parent node is suspected as malicious node and the detected forwarding misbehavior (c_{mis}) is increased by one. When the number of detected forwarding misbehaviors of the suspected node reaches a threshold (φ), the node broadcasts an *Isolate* packet to its one-hop neighbor nodes to prevent the suspected parent node from involving any forwarding operations as shown in Subfig. 2(c). Major operations of the CMD are summarized in Fig. 4.

V. Performance Evaluation

We conduct extensive simulation experiments using the OMNeT++ [6] to evaluate the performance of the proposed approach. 100 nodes are uniformly distributed in a 200×200 m^2 square network area, where a single DODAG root is deployed. The communication range of each node is 30 (m). 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 [16]. To emulate

Notations:

- T^O, φ, r_{los}, r_{n_los}, T_{ch_loss}, t_{cur}, c_{mis}, $FR[rid, fp, uf, ts, \omega]$: Defined before.
- T^{REP}: The timeout period of receiving REP_{plr} packet.
- $pkt[type, seq, src, rec]$: A packet with a sequence number, seq, packet sender, src, and packet receiver, rec. Here, $type$ is *Data*, REQ_{plr}, REP_{plr} or *Isolate*.
- $R_{flag}[seq]$: A $pkt[Data, seq]$ retransmission flag.

◇ When a node n_s generates or receives $pkt[Data, seq]$ destined to the DODAG root:
  Forward $pkt[Data, seq, s, p]$ to n_p; /* n_p is preferred parent node */
  $FR_s[p].fp$ += 1; Set up T^O; /* Eq. 4 */

◇ When a node n_s does not overhear $pkt[Data, seq]$ forwarded by preferred parent node n_p before T^O expires;
  if $R_{flag}[seq]$ is **false**
    Retransmit $pkt[Data, seq, s, p]$ to n_p;
    $FR_s[p].uf$ += 1; $R_{flag}[seq]$ = **true**;
  else
    $FR[p].uf$ += 1; Quit retransmission;

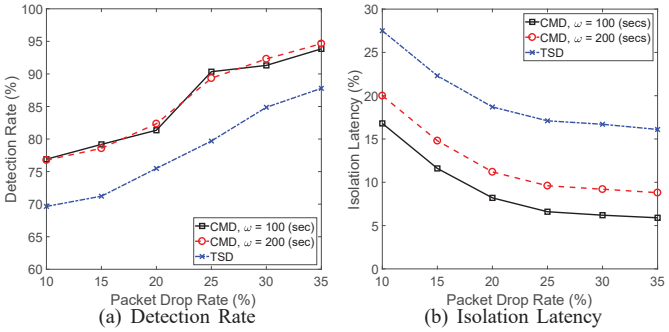
◇ When evaluation window ω ends at node n_s:
  $r_{los} = \frac{FR_s[p].uf}{FR_s[p].fp}$; /* n_p is preferred parent node of n_s */
  Set $FR_s[p].ts = t_{cur}$; /* Set the beginning timestamp of evaluation window to current time */
  if $r_{los} >= T_{ch\_loss}$
    Broadcast $pkt[REQ_{plr}, seq]$;
    Set up T^{REP} /* Eq. 4 */;

◇ When T^{REP} expires at node n_s:
  /* Assume n_s receives N number of REP_{plr} */
  $r_{n\_los} = \frac{\sum_{i=0}^{N} REP^i_{plr}}{N}$;
  if $r_{los} > r_{n\_los}$
    c_{mis} += 1;
    if $c_{mis} > \varphi$
      Broadcast $pkt[Isolate]$;

Fig. 4. The pseudo code of the proposed CMD scheme.

low data rate scenarios, packet injection rate is set to 0.1 pkt/sec. A set of malicious nodes are randomly located in the network. The total simulation time is 3000 seconds. In this paper, we measure the performance in terms of detection rate, isolation latency, packet delivery ratio (PDR), and energy consumption by changing key simulation parameters, including evaluation window (ω), packet drop rate (r_{drop}), and the number of malicious nodes. For performance comparison, we compare the proposed CMD scheme against standard RPL routing protocol without detection mechanism [2] and two-step detection approach (later, *TSD*) [5]. In the TSD, each node observes the communication behaviors of its neighbor nodes by overhearing *Data* packets transmitted by its neighbor nodes and counts any misbehaving activity. If the detected misbehaving activity events exceed the threshold value, the monitoring node suspects the forwarding misbehaviors of the neighbor node and sends the verification packet to the DODAG root to verify whether the sending or forwarding packet was received or not.

First, we measure detection rate and isolation latency by changing r_{drop} and ω in Fig. 5. As r_{drop} increases, both detection rates of CMD and TSD schemes increase in Subfig. 5(a). This is because malicious nodes with larger r_{drop} can have more chances to drop the received *Data* packet and frequently show forwarding misbehaviors. However, these forwarding misbehaviors can be detected by both CMD and TSD. In particular, the CMD scheme shows higher detection rate than that of the TSD scheme. This is because each node



(a) Detection Rate (b) Isolation Latency

Fig. 5. The performance of detection rate and isolation latency against packet drop rate.

monitors the forwarding behaviors of the preferred parent node to observe the packet loss rate, compares the evaluation result with the collected packet loss rate from neighbor nodes, and then detects more forwarding misbehaviors. In the TSD, the monitoring node sends the verification packets (RREQ and RRES) to the DODAG root to verify the forwarding misbehaviors of the preferred parent node. However, verification packets (RREQ and RRES) could get lost due to link failure, hidden terminal problem, or malicious packet drop, the detection process will fail and less number of forwarding misbehaviors can be detected. In addition, evaluation window ω does not show the impact on the detection rate of CMD since the accumulated number of detected forwarding misbehaviors is used to compute the detection rate. In Subfig. 5(b), the overall isolation latency of CMD and TSD decrease as r_{drop} increases. This is because malicious nodes frequently show the forwarding misbehaviors by dropping the received *Data* packets, and more forwarding misbehaviors can be detected within a short time period. The CMD scheme with different ω can achieve much lower isolation latency compared to that of TSD. Unlike our approach, TSD requires the monitoring node to send a verification message to the DODAG root to verify the forwarding misbehaviors of the preferred parent node. Due to malicious packet drop or bad channel quality, the verification messages could get lost and the detection process fails, which would result in higher isolation latency. With a smaller evaluation window ω, lower isolation latency is achieved since the monitoring nodes frequently request the packet loss rate from the one-hop neighbor nodes, more forwarding misbehaviors can be detected within a short time period, and then the malicious nodes can be isolated from the network more quickly.

Second, we measure PDR by varying ω, r_{drop} and the number of malicious nodes in Fig. 6. In Subfig. 6(a), PDR quickly decreases as r_{drop} increases because malicious nodes can have more chances to intentionally drop the received *Data* packets, resulting in the decrement of PDR. The CMD and TSD show higher PDR than that of the original RPL without detection mechanism, this is because forwarding misbehaviors can be detected by both CMD and TSD, malicious nodes can be removed from the network quickly, and more *Data* packets can be delivered to DODAG root. The CMD shows better performance in terms of PDR because the monitoring nodes can quickly retransmit the cached *Data* packet to the

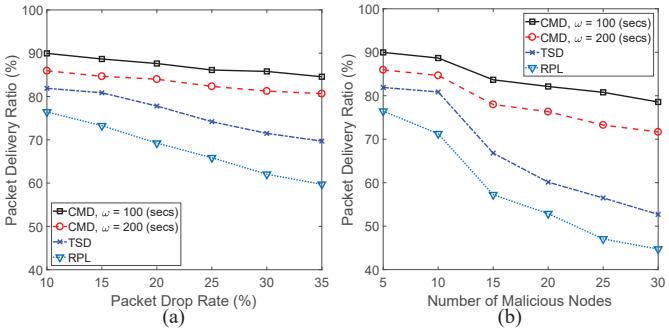


Fig. 6. The performance of packet delivery ratio against packet drop rate and number of malicious nodes.

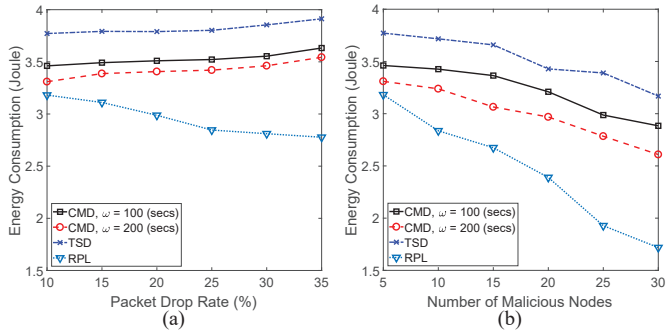


Fig. 7. The performance of energy consumption against packet drop rate and number of malicious nodes.

preferred parent node after the timeout period, more *Data* packets can be delivered to DODAG root. In particular, the CMD with smaller ω shows the best performance since the malicious nodes can be isolated from the network in a shorter time period. Therefore, more *Data* packets can be delivered. As the number of malicious nodes increases in Subfig. 6(b), the overall PDR decreases quickly since more number of *Data* packets can be randomly dropped by more malicious nodes. However, the CMD still shows the best performance and the PDR decreases gracefully compared to that of TSD and original RPL.

Third, the energy consumption is measured based on the number of forwarded and overheard packets [17] by varying ω, r_{drop} and the number of malicious nodes in Fig. 7, where the RPL without detection mechanism shows the lowest energy consumption and is used as the lower bound of the performance. In Subfig. 7(a), the CMD shows lower energy consumption than that of the TSD. This is because each intermediate node in the TSD generates or forwards a large number of control packets (RREQ and RRES) to detect the forwarding misbehavior of the preferred parent node, which consumes more energy than that of CMD. As r_{drop} increases, the energy consumption of CMD and TSD increases. This is because more control packets are generated to detect the forwarding misbehaviors of the malicious nodes, and more energy consumption is observed. In Subfig. 7(b), overall energy consumption decreases as the number of malicious nodes increases in the network. This is because more number of *Data* packets are dropped as the number of malicious nodes increases, the number of dropped *Data* packets is larger than the number of generated and forwarded control packets for the forwarding misbehavior detection, thus overall energy consumption is decreased.

VI. Concluding Remarks

In this paper, we proposed a countermeasure to forwarding misbehaviors in low power and lossy networks running with RPL. The potential forwarding misbehavior of RPL and its performance impact are investigated with a preliminary result. Then, a monitor-based approach, called *CMD*, is proposed to efficiently detect the forwarding misbehaviors of the malicious nodes. Extensive simulation results indicate that the proposed approach achieves better performance, not only improving the detection rate and packet delivery ratio, but also reducing the energy consumption and isolation latency compared to the existing two-step detection approach.

Acknowledgment

This research was supported by Startup grant in the Division of Computer Science at Marshall University.

References

[1] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015.

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

[3] 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.

[4] L. Wallgren, S. Raza, and T. Voigt, "Routing Attacks and Countermeasures in the RPL-based Internet of Things," *Int. J. Distrib. Sens. Netw.*, 2013.

[5] F. Ahmed and Y. Ko, "Mitigation of black hole attacks in Routing Protocol for Low Power and Lossy Networks," *Security and Communication Networks*, vol. 9, pp. 5143–5154, 2016.

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

[7] 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.

[8] A. Le, J. Loo, A. Lasebae, A. Vinel, Y. Chen, and M. Chai, "The Impact of Rank Attack on Network Topology of Routing Protocol for Low-Power and Lossy Networks," *IEEE Sensors Journal*, vol. 11, no. 10, pp. 3685–3692, 2013.

[9] A. Dvir, T. Holczer, and L. Buttyan, "VeRA-version number and rank authentication in rpl," in *Proc. IEEE MASS*, 2011, pp. 709–714.

[10] A. Mayzaud, A. Sehgal, R. Badonnel, I. Chrisment, and J. Schonwalder, "Mitigation of Topological Inconsistency Attacks in RPL-based Low-Power Lossy Networks," *International Journal of Network Management*, vol. 25, no. 5, pp. 320–339, 2015.

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

[12] J. Ren, Y. Zhang, K. Zhang, and X. Shen, "Adaptive and Channel-Aware Detection of Selective Forwarding Attacks in Wireless Sensor Networks," *IEEE Trans. on Wireless Commun.*, vol. 15, no. 5, pp. 3718–3731, 2016.

[13] J. Granjal, E. Monteiro, and J. S. Silva, "Security for the Internet of Things: A Survey of Existing Protocols and Open Research Issues," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 3, pp. 1294–1312, 2015.

[14] W. Stallings, *Cryptography and Network Security - Principles and Practices, 7th Edition*. Pearson, 2016.

[15] M. Petrova, J. Riihijarvi, P. Mahonen, and S. Labella, "Performance Study of IEEE 802.15.4 Using Measurements and Simulations," in *Proc. IEEE WCNC*, 2006, pp. 487–492.

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

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