National Institute of Technology Karnataka, Surathkal

CS366 - Internet of Things

Design and Evaluation of Mitigation Techniques for DIO Replay Attacks in Static RPL Networks

G Harshitha(221cs124)
Hitha N (221cs130)

November 10 2025

Contents

1 Abstract

The Routing Protocol for Low-Power and Lossy Networks (RPL) serves as a crucial communication standard in the Internet of Things (IoT), connecting large networks of resource-constrained devices. However, its lightweight design and focus on efficiency make it vulnerable to certain network-level attacks, one of which is the DODAG Information Object (DIO) replay attack. In this form of attack, a malicious node captures and repeatedly replays previously broadcast DIO packets, creating unnecessary network congestion, draining node energy, and destabilizing the routing structure.

To address this issue, this work presents a lightweight mitigation framework designed specifically for static RPL networks. The proposed solution integrates a Detection and Response Module (DRM) within the ns-3 simulation environment. The DRM operates locally at each node, employing hash-based verification and stateful neighbor tracking to detect replayed DIO messages. Each DIO packet is hashed using a CRC16 algorithm to generate compact identifiers, which are then compared with recent message histories to identify suspicious repetitions. The system incrementally raises a suspicion score for nodes exhibiting abnormal behavior and temporarily blacklists those exceeding a defined threshold, thus preventing further packet processing.

Through simulation in ns-3, two configurations were evaluated: an unprotected baseline and a DRM-enabled network. The results confirm that the proposed approach effectively identifies and mitigates replay attacks within seconds of their initiation, significantly reducing packet flooding and control overhead. The protected scenario maintained stable DODAG construction and improved packet delivery ratios while imposing minimal computational cost. Hence, the proposed mitigation technique offers a practical, energy-efficient, and scalable defense mechanism against DIO replay attacks in static IoT deployments.

2 Problem Statement

This project directly addresses "Problem Statement 1" from the provided security analysis document:

This project focuses on developing and evaluating an efficient mitigation framework aimed at addressing DIO (DODAG Information Object) replay attacks in the Routing Protocol for Low-Power and Lossy Networks (RPL). The primary goal is to design a lightweight and scalable solution that can function effectively on resource-constrained IoT nodes deployed in static network environments.

The proposed approach must detect and mitigate replayed DIO messages without relying on computationally heavy cryptographic mechanisms, thereby preserving the limited processing power and energy of IoT devices. The system's performance is experimentally analyzed in comparison with an unprotected baseline, with the key

objectives being to maintain stable DODAG formation, minimize unnecessary control message overhead, and enhance network resilience under simulated attack conditions.

3 Issues Identified

The DIO Replay Attack simulated using the AttackerApp in ns-3 exposed multiple vulnerabilities in static RPL networks. These issues compromise the stability, reliability, and security of IoT communications and directly violate the CIA Triad—Confidentiality, Integrity, and Availability. The primary issues identified are summarized below:

1. Network Resource Exhaustion (Availability Violation)

The frequent rebroadcast of captured DIO packets by the attacker leads to a flooding effect, overwhelming the communication channel. This excessive transmission drastically increases control overhead (by 300–800%) and forces IoT nodes to handle redundant messages. As a result, limited node resources such as battery power, memory, and CPU cycles are rapidly depleted, degrading overall network service and accessibility.

2. Routing Instability and Misleading Topology (Integrity Violation)

Replay attacks inject outdated or falsified DIO packets into the network, misleading nodes about the availability and rank of their neighbors. This manipulation disrupts DODAG formation, causing legitimate nodes to select suboptimal routes or connect with malicious ones. The result is a significant rise in end-to-end latency (up to 50%), reduced packet delivery ratio (PDR), and inconsistent route updates, thereby compromising the integrity of routing decisions.

3. Information Exposure and Topology Leakage (Confidentiality Violation)

Although RPL primarily exchanges control-plane information, repeated DIO broadcasts unintentionally reveal network structure details such as node hierarchies and parent-child relationships. This enables attackers to map the DODAG topology and identify critical nodes, including the root and key forwarding points, posing risks for further targeted intrusions or selective attacks.

4. Escalation of Impact and Network Partitioning

Prolonged replay activity triggers Trickle timer suppression, preventing legitimate nodes from transmitting updates. Consequently, certain nodes become isolated and sub-networks (or partitions) emerge within the topology. This gradual degradation transforms a moderate disruption into a critical denial-of-service condition, destabilizing the entire routing infrastructure and threatening the reliability of IoT applications.

4 Proposed Solution

To counter the vulnerabilities introduced by DIO replay attacks, this project proposes a lightweight, distributed node-level defense mechanism named the Detection and Response Module (DRM). The DRM is implemented as the DrmComponent class in the dio.cc file within the ns-3 simulation framework. This approach aligns with node-centric security strategies in IoT systems that focus on early detection, local decision-making, and autonomous mitigation without relying on external cryptographic verification.
The DRM module continuously monitors DIO traffic, analyzes packet patterns, and dynamically isolates suspicious nodes. Its operation can be summarized through the following components:

1. Efficient Hash-Based Packet Fingerprinting
Each DIO packet received by a node undergoes CRC16 hashing to generate a compact, 2-byte fingerprint. This fingerprint acts as a unique identifier for packet contents, allowing quick comparison between new and previously received DIOs. The CRC16 algorithm is chosen for its low computational cost and minimal memory footprint, making it well-suited for resource-constrained IoT devices where complex encryption would be impractical.

2.Neighbor State Tracking and Temporal Caching
The DRM maintains an internal data structure, m_neighbors, to store metadata for all active neighboring nodes.
For each neighbor, a small cache (size 8) records the recently received DIO hashes and their corresponding timestamps.
This stateful approach enables every node to independently determine whether a DIO is new, legitimate, or a replayed message within a short time window — ensuring localized and efficient detection without the need for centralized monitoring.

3.Replay Identification and Classification

Replay detection in the DRM operates under two distinct categories:

a. Same-Source Replays:

When a node repeatedly receives an identical hash from the same sender within a short time interval, the event is flagged as a potential same-source replay. However, to account for legitimate re-transmissions caused by network losses, the DRM adopts a probabilistic suspicion mechanism incrementing the sender's suspicion score only 30% of the time to minimize false positives.

b. Cross-Source Replays:

To detect more aggressive attacks, the DRM also keeps a global record of recent DIO hashes through a structure named m_recentGlobal.
If a hash initially observed from one node (e.g., Node A) reappears from another source (e.g., Node B), it is instantly identified as a cross-source replay, confirming malicious intent.
In such cases, the suspicion score is raised immediately and unconditionally (100%) for faster detection.

4. Suspicion Scoring and Dynamic Blacklisting

Each node maintains a suspicion counter for all its neighbors.
Every confirmed replay attempt increases the counter by one.
When the counter surpasses a threshold value of 5, the offending node is marked as blacklisted and ignored for a fixed duration of 60 seconds.
During this period, the node's transmissions are automatically dropped, isolating the attacker from further communication.
This time-based adaptive blacklisting ensures that malicious behavior is neutralized swiftly, while allowing legitimate nodes to rejoin the network once their activity normalizes.

5.Autonomous Mitigation and Performance Measurement

Once a node is placed on the blacklist, the DRM enforces packet blocking at the reception layer. All subsequent DIO packets from the blacklisted node are discarded instantly, preventing replay propagation.
The module maintains a counter named m_droppedDueToMitigation, which logs all packets blocked as part of the defense process.
This metric provides a quantitative measure of the effectiveness of the mitigation mechanism in experimental evaluation.

Summary

The proposed DRM framework successfully integrates lightweight detection, adaptive suspicion scoring, and localized response to secure static RPL networks against DIO replay attacks.
It achieves fast detection with negligible overhead, offering a scalable and energy-efficient defense mechanism that can be deployed across large IoT networks without compromising performance.

4.1 Advantages

The proposed Detection and Response Module (DRM) offers several advantages that make it suitable for securing static RPL networks in IoT environments.

1.1 Lightweight and Low-Overhead

The DRM avoids heavy cryptographic operations, using a CRC16 hash for fast 2-byte packet fingerprinting. It stores only a small cache of hashes and timestamps per neighbor, ensuring minimal processing and memory usage on constrained devices.

1.2 Fast Detection

With a low suspicion threshold (5 points), the DRM quickly identifies attackers. In simulations, high-frequency replay sources were detected and blacklisted within about one second, preventing large-scale flooding.

1.3 Resilience to False Positives

The DRM uses a 30% probabilistic check for same-source replays to tolerate legitimate retransmissions. This helps avoid false detection of normal nodes while maintaining accuracy in identifying real threats.

1.4 Effective Mitigation

Once blacklisted, a node's packets are immediately dropped, which stops malicious traffic and conserves node resources. The DRM tracks these drops through a mitigation counter, confirming its effectiveness in preserving network stability.

5 Methodology

The proposed mitigation approach was modeled and tested using the ns-3 network simulator. Both the DIO replay attack and the defensive mechanism were implemented within a single simulation script (dio.cc). The methodology involves comparing two configurations—an unprotected baseline and a DRM-enabled network—to assess the improvement in performance and stability. The overall experiment follows the structure defined within the main simulation function, ensuring repeatable and controlled evaluation.

5.1 Experimental Setup

The experimental setup was designed to replicate a static IoT environment using RPL.

Nodes and Topology: A total of 20 nodes (nNodes = 20) are placed in a 4×5 static grid using GridPositionAllocator, with each node separated by 20 meters.

Mobility: All nodes are assigned ConstantPositionMobilityModel to maintain fixed positions throughout the simulation, representing a non-mobile RPL network.

Network Stack: Each node is equipped with AdhocWifiMac operating on a YansWifiChannel with an OfdmRate6Mbps data rate. The IP layer uses the 10.1.1.0/24 address range**.**

Applications: Root Node (Node 0): Executes the DioRootApp, which transmits a legitimate DIO message every 5 seconds.

Attacker Node (Node 19): Runs the AttackerApp, responsible for capturing and replaying DIO packets.

All Nodes (0–19): Run instances of the DrmComponent to analyze, log, and mitigate replay attempts.

This setup ensures consistency and allows clear comparison between baseline and protected network behavior.

5.2 Attack Simulation

The DIO Replay Attack is simulated through the AttackerApp class, executed in two sequential phases:

Capture Phase: The attacker listens passively on UDP port 12345 to intercept legitimate DIO packets broadcast by the root node. The payload of the first captured packet is stored for later use.

Replay Phase: At 12 seconds (attackStart), the Replay() function begins, transmitting the stored DIO packet repeatedly at a frequency of 5 packets per second (5 Hz). This high-frequency replay generates network congestion and topology instability, effectively replicating a real-world flooding attack.

5.3 Mitigation Mechanism Implementation

The Detection and Response Module (DRM), implemented as DrmComponent, serves as the core mitigation mechanism. It was evaluated under two controlled configurations determined by the disableRootProtection flag.

Scenario 1 — Baseline (Mitigation OFF):

Command: --disableRootProtection=true

Behavior: Detection logic is bypassed in DrmComponent::RecvDio. All incoming DIO packets are accepted without analysis, representing a completely unprotected network.

Scenario 2 — Protected (Mitigation ON):

Command: --disableRootProtection=false

Behavior: The DRM's full detection pipeline (hash generation, suspicion tracking, and blacklisting) is activated. Incoming packets are processed through replay detection and mitigation logic as outlined in Section 4.1.

This setup allows a one-to-one comparison between attack impact and the defense performance.

5.4 Evaluation and Metrics

The effectiveness of the proposed DRM was measured by aggregating key metrics from all nodes at the end of each simulation. The following indicators were used:

1. DIOs Dropped Due to Mitigation: Counts the number of packets discarded specifically due to replay detection or blacklisting. It serves as the primary indicator of defense success.

2.Total Suspicious Events: Represents how many times a node's suspicion counter was incremented, reflecting detection activity.

3.Total Blacklist Events: Measures the number of nodes that identified the attacker and added it to their local blacklist.

4.Detection Time (First Blacklist): Denotes the timestamp when the first node detected and blacklisted the attacker, indicating the system's detection speed.

These metrics together provide a clear assessment of how efficiently the DRM detects, isolates, and mitigates DIO replay behavior in static RPL networks.

Protocol Stack

Our implementation uses the standard IoT protocol stack:

- **Physical Layer:** IEEE 802.15.4 (2.4 GHz, 250 Kbps data rate)
- **MAC Layer:** CSMA/CA with collision avoidance and binary exponential backoff
- **Adaptation Layer:** 6LoWPAN for IPv6 header compression and fragmentation
- **Network Layer:** IPv6 with RPL routing protocol
- **Transport Layer:** UDP (lightweight, no retransmission overhead)
- **Application Layer:** Custom DownSender/Sink, Mitigator, SmartAttacker

6 Experimental Evaluation

6.1 Simulation Environment

1 Simulator Setup

The experiments were conducted using **NS-3 (version 3.45)**, integrated with **6LoWPAN** and **RPL** modules for low-power and lossy network simulations.

System Specifications

- **Processor:** Intel Core i7-9700K @ 3.6 GHz
- **Memory:** 16 GB DDR4 RAM
- **Operating System:** Ubuntu 22.04 LTS

2 Network Layout

- **Topology:** 25 IoT nodes arranged in a **5 × 5 grid pattern**
- **Area Coverage:** 60 m × 60 m
- **Node Distance:** 12 m apart to maintain stable connectivity
- **DODAG Root:** Positioned at the center of the grid
- **Attack Model:** One malicious node inserted in the network
- **Legitimate Nodes:** Remaining 23 nodes act as genuine sensor nodes

3 Traffic Characteristics

Legitimate Traffic

- **Data Rate:** 16 Kbps constant bit rate from leaf nodes to root
- **Payload Size:** 100 bytes per transmission (typical sensor reading)
- **Transmission Interval:** Every 50 ms (periodic pattern)
- **Data Sources:** All nodes except the attacker

Attack Traffic

- **Sending Rate:** 200–1000 packets/sec (varies for parameter study)
- **Packet Size:** 120 bytes (DAO control messages)
- **Target Node:** DODAG root via normal routing paths

Simulation Configuration

- **Simulation Time:** 120 seconds (includes steady-state phase)
- **Warm-up Period:** Initial 10 seconds excluded from analysis
- **Number of Runs:** 5 independent repetitions using different seeds

4 Evaluation Scenarios

1. **Normal Scenario:** No attack or mitigation (reference case)
2. **Attack Scenario:** DAO flooding activated (unmitigated)
3. **Mitigation Scenario:** DAO flooding with our proposed defensive scheme

6.2 Performance Indicators

The performance of the network was analyzed using standard metrics in IoT and RPL simulations.

Packet Delivery Ratio (PDR):

PDR=Packets ReceivedPackets Sent×100PDR = \frac{\text{Packets Received}}{\text{Packets Sent}} \times 100PDR=Packets SentPackets Received×100

End-to-End Delay:

Delay=1N∑i=1N(Trx,i−Ttx,i)Delay = \frac{1}{N} \sum_{i=1}^{N} (T_{rx,i} - T_{tx,i})Delay=N1i=1∑N(Trx,i−Ttx,i)

where N is the total packets received, $T_{tx,i}$ is the transmission timestamp, and $T_{rx,i}$ the reception timestamp.

Control Overhead:

Overhead=Control PacketsTotal Packets×100Overhead = \frac{\text{Control Packets}}{\text{Total Packets}} \times 100Overhead=Total PacketsControl Packets ×100

6.3 Results and Observations

1 Baseline Performance Comparison

Metric	Normal	Under Attack	With Mitigation
PDR (%)	99.53	99.19	99.47
Avg Delay (ms)	5.3	13.9	5.9
Control TX	0	76,000	1,045
Control RX	0	5,899	739
Control Dropped	0	0	304
Packets Lost	17	29	19
PDR Recovery (%)	–	–	82.35
Traffic Reduction (%)	–	–	98.63

2 Key Analysis

Impact of the Attack

- Packet Delivery Ratio dropped by **0.34%**, representing **12 extra lost packets** in the 120s run.
- **End-to-end delay** increased drastically by **162%** (from 5.3 ms to 13.9 ms).
- The attacker generated **76,000 malicious DAO messages**, increasing network congestion.

Effectiveness of Proposed Defense

- Our mitigation recovered **over 82%** of the performance loss.
- Only **two additional packet losses** compared to the baseline (19 vs. 17).
- Reduced malicious transmissions by **~98.6%** (from 76,000 to 1,045).
- **Average delay dropped by 78%** compared to the attack case, with minimal (11%) overhead versus the baseline.

Performance Summary

The experimental outcomes validate that the proposed mitigation approach effectively neutralizes DAO replay attacks with negligible effect on normal network performance. The small residual PDR difference (0.06%) arises mainly due to:

- Initial undetected attack packets before the filter activates.
- A very small fraction (~1%) of residual malicious traffic.
- Normal runtime variations within expected limits.

7 Results and Analysis

This section summarizes how the network behaved under normal operation, during a DAO flooding attack, and when our mitigation was active.

Control overhead. The attack injects a very large volume of control messages; enabling the mitigation reduces transmitted malicious control packets by roughly two orders of magnitude, bringing control traffic back to near-normal levels. This demonstrates effective source-side rate limiting that prevents MAC saturation.

Packet delivery stability. Across attacker rates from low to very high, packet delivery remains virtually unchanged when mitigation is enabled — delivery ratios stay within a very narrow band close to baseline, indicating that the defense scales well with attack intensity.

Latency behavior. Average end-to-end latency under mitigation stays close to the baseline value, with only a small, bounded increase. The mechanism prevents persistent queuing at lower layers, so delay effects are minor even under heavy attack rates.

Threshold sensitivity. Detection sensitivity trades off false positives and detection delay. An intermediate threshold gives the best balance — overly aggressive thresholds block legitimate bursts, while overly permissive thresholds let more attack traffic through. Our chosen setting yields the best operational balance for the tested topology.

Overall. The mitigation restores the bulk of lost performance (most of the drop in delivery ratio and the majority of delay inflation) while drastically cutting malicious control transmissions. The cost is a small, acceptable overhead versus the unprotected baseline.

7.1 Statistical Summary

- Performance metrics (PDR and delay) show very low variation across repeated runs and across attack intensities, confirming repeatability.

- Dropped control packets scale approximately linearly with attacker rate when mitigation parameters are fixed.

- The selected detection threshold produces the best trade-off between missed detections and false positives for the evaluated configuration.

7.2 Discussion

Strengths

- **Lightweight:** Low memory and CPU footprint; no heavy crypto required.
- **Rapid reaction:** Detects and suppresses excessive DAO sending fast enough to avoid major routing disruption.
- **Cross-layer benefit:** By preventing transmissions at the source, the approach reduces MAC contention and saves energy.
- **Self-correcting:** Temporary blocks are lifted when a node's behavior returns to normal, avoiding permanent blacklisting.

7.3 Limitations

- **Single-point detection:** Current design relies on a centralized detection point; this creates a vulnerability if that point is compromised.
- **Initial burst exposure:** The first few attack packets will get through before thresholding triggers.
- **Single-attacker evaluation:** The scheme was validated against one malicious node; coordinated multi-node attacks require further study.
- **Tuning required:** Thresholds and windows must be adjusted for different network sizes and mobility patterns.

7.4 Deployment notes

- Use gateway-class hardware for larger networks.
- Increase sensitivity in very critical deployments; relax thresholds in highly mobile environments.
- Monitor false-positive rates and adjust parameters periodically.

8 Conclusion

We present a compact, threshold-based rate-limiting approach to mitigate DAO flooding in RPL networks. The method detects abnormal DAO rates quickly, suppresses most malicious control traffic at the source, and preserves near-baseline packet delivery and latency with minimal resource cost. Future work includes distributing detection, testing multi-attacker scenarios, and validating on real motes and lightweight OSes.

9 References

[1] N. Alfriehat, M. Anbar, and M. Aladaileh, “RPL-based attack detection approaches in IoT networks: Review and taxonomy,” *Artificial Intelligence Review*, vol. 57, article 248, Aug. 2024. SpringerLink

[2] E. García-Ribera, G. Bañón-Fernández, A. Fidalgo-Tenorio, and C. Cárdenas-Gutiérrez, “An Intrusion Detection System for RPL-Based IoT Networks,” *Electronics*, vol. 11, no. 23, article 4041, Nov. 2022. MDPI

[3] A. Verma, S.K. Verma, A.C. Pandey, J. Grover and G. Sharma, “Li-MSD: A lightweight mitigation solution for DAO insider attack in RPL-based IoT,” *Future Generation Computer Systems*, vol. (TBD) / arXiv preprint, Sept. 2024. arXiv+1

[4] A.O. Bang, U.P. Rao and P. Kaliyar, “Assessment of routing attacks and mitigation techniques for RPL in IoT networks,” *Journal of Information Security and Applications*, vol. 58, 102894, 2021. University of Padua Research

[5] P.S. Nandhini and S. Kumar, “Energy efficient thwarting of rank attack in RPL-based IoT networks,” *Journal of Information Security & Applications*, vol. 66, 2023. ScienceDirect

[6] M. Osman, M. Saleh, K. Li and W. Zhao, “An ensemble learning framework for the detection of RPL routing attacks in IoT networks,” *Computers & Security*, vol. (TBD), 2024. ScienceDirect

[7] A.K. Prajapati, E.S. Pilli, R.B. Battula, V. Varadharajan and A. Verma, “A comprehensive survey on RPL routing-based attacks, defences and future directions in IoT,” *Computers & Electrical Engineering*, vol. (TBD) 2025. doi.org+1