DOI: https://doi.org/10.57233/ihcfs.016

ISBN: 978-978-68-07775

BOOK OF PROCEEDINGS

IHCFS – 2025 WEBSITE: fos.fugusau.edu.ng

1ST INTERNATIONAL HYBRID CONFERENCE OF FACULTY OF SCIENCE

(A PUBLICATION OF FACULTY OF SCIENCE, FEDERAL UNIVERSITY GUSAU, NIGERIA)

Static vs Roaming Hatchetman: Lightweight Performance Metrics for Mixed-Mobility IoT Meshes

Jameel Shehu Yalli[1*], Joshua Teddy Ibibo[2], Peter W. Odey[3], Abdulrahman Suleiman Ogah[4], Ilodigwe Precious Adaeze[5], Yahaya Shuaibu Musa[6], Faisal Ali Garba[7]

Department of Computer Science, Federal University Gusau, Zamfara[1]

School of Computing, Engineering & Digital Technologies, Teesside University, Middlesbrough, United Kingdom[2]

Centre for Cyberspace Studies, Nasarawa State University Keffi[3]

Nigeria Security and Civil Defence Corps[4]

Byte Bunker Limited, Abuja[5]

Yahmis Global Limited, Kano[6]

Phoenyx Security, Kano[7]

***Corresponding Author:alifa2try@gmail.com**

Abstract

The Routing Protocol for Low-Power and Lossy Networks (RPL) in non-storing mode is vulnerable to a security threat called the Hatchetman attack, which targets downward traffic. Instead of dropping packets directly, the attacker secretly modifies the Source Routing Header (SRH) — the list of next hops a packet must follow. Once altered, the packet is later lost or misrouted without visible warning, allowing the attacker to appear well-behaved. This makes the attack difficult to detect. Most past studies evaluate Hatchetman only in either static networks or fully mobile networks, which does not reflect how real Internet-of-Things (IoT) systems usually operate. Practical deployments are hybrid, meaning some nodes move and others remain fixed. When an attacker also moves, the risk changes, because movement increases the chance of being in the packet path. To study this clearly, we built a lightweight SimPy simulation model where each packet carries its own SRH, and attacker nodes include a hook that edits the remaining hops. The attack begins after a short warm-up period to capture baseline behavior. We test three scenarios across ten runs: (1) a static network, (2) a hybrid network with a fixed attacker, and (3) a hybrid network with a roaming attacker. We measure Packet Delivery Ratio (PDR), Average End-to-End Delay (AE2ED), control-plane airtime, energy usage (proxy), throughput, and error messages. Baseline networks deliver all packets (PDR = 1.00). Under attack, PDR drops to about 0.44 in static mode, 0.78 when the hybrid attacker is static, and 0.69 when the hybrid attacker is roaming. Control-plane airtime stays near 44–49%, and delay lowers slightly because only faster packets succeed. Roaming attackers do the most damage by reaching more routes over time. These results highlight the need for mobility-aware monitoring and lightweight SRH integrity checks to prevent hidden manipulation in real IoT systems.

Keywords: RPL, IoT, DODAG, SimPy, Simulation

1.0 INTRODUCTION

The Internet Protocol version 6 (IPv6) Routing Protocol for Low-Power and Lossy Networks (RPL) is the de facto routing standard for resource-constrained IoT systems. It builds a Destination Oriented Directed Acyclic Graph (DODAG), where each node computes a rank indicating its position relative to the root, enabling loop-free parent selection through metrics such as hop count or ETX (Alfriehat, 2024). Although originally optimized for upward traffic, RPL also supports downward communication—critical for applications like smart homes, healthcare monitoring, and industrial automation—where control messages must

DOI: https://doi.org/10.57233/ihcfs.016

ISBN: 978-978-68-07775

reliably reach nodes (Alfriehat, 2024). Downward reliability directly affects security, since compromised routes can result in dropped packets, failed commands, or breaches in sensitive environments. Attacks such as blackhole, sinkhole, and selective forwarding highlight the need for secure downward routing to maintain trust, reliability, and timely response in IoT deployments (Alfriehat, 2024).

RPL operates in storing and non-storing modes. In non-storing mode, the root embeds the entire downward route into a Source Routing Header (SRH), reducing memory overhead at intermediate nodes but creating a key vulnerability: any forwarder can manipulate the SRH. Hatchetman exploits this by modifying SRH entries in-path, replacing valid hops with bogus addresses to silently misroute or drop packets without appearing malicious (Sharma *et al*., 2023; Sharma *et al*., 2022). Unlike traditional forwarding-drop attacks, Hatchetman is stealthier because it alters routing instructions rather than packet handling.

Sharma *et al*. (2023) evaluated Hatchetman under static networks, while Stephen *et al*. (2021) explored downward manipulation under mobility but without SRH-specific focus. Broader RPLsecurity studies (Pu *et al*., 2018; Aydin *et al*., 2024; Mali *et al*., 2021; Almutairi *et al*., 2024; Zilberman *et al*., 2025) emphasize rank and general routing threats, not in-path SRH rewriting.

Three gaps persist:i) lack of hybrid-mobility evaluation (i) no comparison of static vs. roaming attackers in hybridsiii) narrow performance metrics—often limited to PDR/delay—without control-plane airtime, energy, or ETX-based impact assessment.

The objectives of this paper are thus:

a. To develop a lightweight and reproducible SimPy-based simulation model that accurately represents RPL non-storing downward routing and the Hatchetman attack through per-packet SRH planning and in-path tampering hooks.
b. To evaluate the impact of the Hatchetman attack across multiple mobility regimes—a fully static topology (S0), a hybrid topology with a static attacker (H-S), and a hybrid topology with a roaming attacker (H-M)—using consistent experimental conditions and ten-seed runs.
c. To measure the effect of attacker behaviour on key performance metrics, including Packet Delivery Ratio (PDR), Average End-to-End Delay (AE2ED), control-plane airtime, energy consumption, throughput, error packets, and attack occurrences.
d. To determine how attacker mobility affects disruption severity**,** specifically comparing static vs. roaming adversaries within hybrid (mixed-mobility) IoT networks.

2.0 MATERIALS AND METHODS

The study uses a lightweight SimPy-based simulator to model RPL non-storing downward routing and Hatchetman-style SRH manipulation. A single simpy.Environment() hosts NUM_NODES placed uniformly in a 2D area with minimum spacing; node creation is paced so time advances during bootstrap. Neighbor relationships are based on Euclidean distance ≤ REACH_RADIUS, and nodes refresh neighbor lists continuously. Each node runs three background processes: an inbox handler, a Trickle-like DIO timer, and a small link-churn routine that randomly drops neighbor edges.

For mobility (hybrid runs), non-root nodes become mobile with probability MOBILE_FRACTION. Every simulated second, mobile nodes perform a random-waypoint step (speed from MOBILE_SPEED_RANGE, optional pauses), clamping positions to the area and refreshing neighbors.

Downward traffic is root-only: every TRAFFIC_INTERVAL, the root selects a destination and computes a hop-by-hop path using BFS. This path becomes a simplified SRH-like route plan embedded in each

DOI: https://doi.org/10.57233/ihcfs.016

ISBN: 978-978-68-07775

BOOK OF PROCEEDINGS

packet. Forwarders follow the next hop in the plan; if topology invalidates it, they fallback to a random neighbor.

The Hatchetman attack model attaches per-packet hooks to selected malicious nodes. The hook intercepts in-transit data and alters the remaining route plan by inserting or replacing the next hop with a fake address, causing downstream drops while preserving the attacker's local forwarding behavior. When a node encounters a fake hop, it drops the packet and emits an ErrorPacket, counted as control overhead.

Two scenarios are supported: static and hybrid mobility, though hybrid runs do not explicitly separate static vs. roaming attackers. Metrics include PDR, AE2ED, control-plane airtime, an energy proxy, throughput, error counts, and attack invocations. Key limitations include simplified downward routing (BFS instead of DAO state), immediate attack activation, and non-enforced attacker mobility classes.

3.0 RESULTS AND DISCUSSIONS

This section reports the measured impact of Hatchetman across three settings—S0 (Static), H-S (Hybrid with a static attacker), and H-M (Hybrid with a roaming attacker)—using 10-seed as provided in Tables 1–5. We highlight reliability (PDR), latency (AE2ED), control-plane share, energy, and error activity, then synthesize the cross-scenario trends and discuss operational implications.

Static Scenario (S0):

The static baseline is consistently reliable: PDR = 1.00 as seen in Table 1 for all seeds. AE2ED is low (~0.11–0.34 s), consistent with short, stable paths; control-plane airtime hovers around ~47–49.5%, dominated by periodic DIOs rather than data or error traffic. Energy follows control activity (≈1.6k–5.0k "units") and throughput reflects the low-rate downstream load.

Table 1: Reference Simulation Static Scenario

Metric	Seed =1	Seed = 2	Seed = 3	Seed = 4	Seed = 5	Seed = 6	Seed = 7	Seed = 8	Seed = 9	Seed = 10
PDR	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
AE2ED	0.1867	0.2050	0.3400	0.2282	0.1081	0.1393	0.1286	0.1960	0.1542	0.1281
Control Packets	2327	1057	1289	1754	1658	1548	1873	2046	1664	3284
Error Packets	0	0	0	0	0	0	0	0	0	0
Control-plane Airtime (%)	48.36	49.50	48.16	47.37	48.37	48.05	48.99	48.88	48.22	49.22
Energy Consumption (units)	3579.00	1597.50	1989.00	2740.50	2550.00	2392.50	2853.00	3121.50	2565.00	4984.50
Throughput (bits/s)	2.99	0.28	0.85	2.42	2.28	1.99	1.00	1.42	1.71	2.28
Attacks Applied	0	0	0	0	0	0	0	0	0	0

Enabling Hatchetman reduces reliability: PDR ranges 0.111–0.833 across seeds, and Error Packets as well as Attacks Applied rise correspondingly. AE2ED decreases slightly on average—an expected survivorship effect (slower, more fragile paths are more likely to be dropped; the delivered subset skews faster). Control-plane share stays in a similar band (≈38–49%), because DIOs dominate the numerator; the additional error ICMPs are modest relative to steady control chatter. In a fixed mesh, Hatchetman meaningfully suppresses delivery when an attacker lies on the route. Seeds where the attacker never becomes in-path show Attacks Applied = 0

DOI: https://doi.org/10.57233/ihcfs.016
ISBN: 978-978-68-07775
BOOK OF PROCEEDINGS
IHCFS – 2025 WEBSITE: fos.fugusau.edu.ng

and baseline-like metrics—an expected negative control.

Table 2: Static Scenario Attack Simulation

Metric	Seed =1	Seed = 2	Seed = 3	Seed = 4	Seed = 5	Seed = 6	Seed = 7	Seed = 8	Seed = 9	Seed = 10
PDR	0.111	0.200	0.364	0.680	0.444	0.833	0.667	0.429	0.600	0.111
AE2ED	0.1233	0.0900	0.1550	0.1082	0.1800	0.2240	0.0900	0.1300	0.1244	0.1233
Control Packets	3606	1155	1428	2156	2163	1623	1894	1789	1685	3606
Error Packets	24	4	7	8	10	1	2	4	6	24
Control-plane Airtime (%)	42.87	42.54	38.32	47.04	45.37	49.04	47.84	48.52	46.41	42.87
Energy Consumption (units)	6336.00	2039.50	2797.00	3426.50	3569.50	2473.00	2964.50	2765.50	2716.50	6336.00
Throughput (bits/s)	0.43	0.14	0.57	47.04	1.14	0.71	0.57	0.43	1.28	0.43
Attacks Applied	24	5	7	8	15	1	2	6	11	24

Hybrid with static attacker (H-S):
With ~50% of nodes mobile, the network maintains PDR = 1.00 across seeds in the hybrid network reference simulation as seen in Table 3, while AE2ED remains small (mostly ~0.13–0.25s, with one long-tail seed at 0.78s due to transient churn). Absolute control traffic and energy increase by an order of magnitude versus S0, but control-plane airtime remains near~48–49%, reflecting that *both* control and data bytes scale up.

Table 3: Hybrid Reference Simulation

Metric	Seed =1	Seed = 2	Seed = 3	Seed = 4	Seed = 5	Seed = 6	Seed = 7	Seed = 8	Seed = 9	Seed = 10
PDR	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
AE2ED	0.2076	0.2235	0.2363	0.2232	0.2450	0.1929	0.1625	0.7771	0.1522	0.1304
Control Packets	35340	23604	26468	30454	30862	27001	35269	31203	32246	43884
Error Packets	0	0	0	0	0	0	0	0	0	0
Control-plane Airtime (%)	49.26	48.23	48.82	48.78	48.58	48.78	49.21	46.33	49.18	49.55
Energy Consumption (units)	53610.00	36381.00	40419.00	46540.50	47308.50	41260.50	53538.00	49585.50	48970.50	66273.00
Throughput (bits/s)	16.92	16.92	16.92	16.92	16.92	16.92	16.92	16.92	16.92	16.92
Attacks Applied	0	0	0	0	0	0	0	0	0	0

A static attacker in a hybrid mesh remains effective: PDR typically falls into the 0.345–0.958 range, Error Packets rise into the tens per run, and throughput drops accordingly. AE2ED again trends slightly lower than the reference (survivorship), while the control-

DOI: https://doi.org/10.57233/ihcfs.016

ISBN: 978-978-68-07775

BOOK OF PROCEEDINGS

plane share changes by only a few tenths of a percentage point because DIO volume dominates.Even without moving, the attacker intersects more paths over time as mobility reshapes routes, producing sustained PDR penalties relative to the hybrid baseline.

Table 4: Hybrid with Static Attackers

Metric	Seed =1	Seed = 2	Seed = 3	Seed = 4	Seed = 5	Seed = 6	Seed = 7	Seed = 8	Seed = 9	Seed = 10
PDR	0.849	0.664	0.706	0.840	0. 345	0.958	0.857	0.857	0.916	0.773
AE2ED	0.1493	0.1197	0.1990	0.1635	0.1832	0.1817	0.1140	0.1939	0.1147	0.1442
Control Packets	41982	26793	26826	29634	27966	31412	38322	30660	34709	37826
Error Packets	18	40	35	19	78	5	17	17	10	27
Control-plane Airtime (%)	48.8	42.63	46.60	47.80	44.55	48.99	48.64	47.62	49.01	48.66
Energy Consumption (units)	64944.00	47014.00	43061.00	46309.00	47089.50	47882.00	58943	48084.50	52987.00	58210.50
Throughput (bits/s)	14.36	11.24	11.95	14.22	5.83	16.21	14.51	14.51	15.50	13.08
Attacks Applied	25	51	63	30	83	6	27	31	17	37

Hybrid with roaming attacker (H-M):
Allowing the attacker to roam amplifies the effect:

- PDR is lower on average than in H-S (typical values ~0.42–0.78), reflecting more frequent on-path placement.
- Error Packets increase (dozens per run), consistent with more injected bogus hops being encountered.
- Throughput is depressed relative to H-S, mirroring the larger PDR loss.
- AE2ED remains low (slightly above H-S in some seeds), again shaped by the delivered subset.

Roaming multiplies the attacker's route-intersection rate, yielding more SRH tampering events and systematically lower delivery than with a fixed attacker.

Table 5: Hybrid with roaming attackers

Metric	Seed =1	Seed = 2	Seed = 3	Seed = 4	Seed = 5	Seed = 6	Seed = 7	Seed = 8	Seed = 9	Seed = 10
PDR	0.420	0.681	0.782	0.782	0.731	0.521	0.773	0.681	0.756	0.773
AE2ED	0.1662	0.1563	0.2056	0.1962	0.2048	0.1492	0.1227	0.1985	0.1181	0.1223
Control Packets	36845	26576	23747	31530	30197	28063	32396	30406	32298	38082
Error Packets	69	38	26	26	32	57	27	37	29	27
Control-plane Airtime (%)	43.85	43.84	46.76	46.42	35.92	38.01	46.50	42.87	48.19	49.03
Energy Consumption (units)	62965.50	45324.50	37959.50	50739.50	62897.00	55300.50	52110.00	53045	50175.50	58179.00
Throughput (bits/s)	7.11	11.52	13.23	13.23	12. 37	8.82	13.08	11.24	12.80	13.08

Attacks Applied	76	58	44	45	46	62	47	53	32	29

Across scenarios, baseline reliability is perfect (PDR = 1.00), but under attack PDR falls least in the static case (S0), more in hybrid with a static attacker (H-S), and most in hybrid with a roaming attacker (H-M), reflecting the increased likelihood of route intersection as topology evolves. Latency slightly decreases in all attack settings due to survivorship bias, where slower or fragile packets are preferentially lost. Control-plane airtime remains stable at ~44–49%, dominated by DIO traffic, with Hatchetman-triggered error packets contributing only a small share. Energy consumption rises from static to hybrid runs because mobility drives continuous radio activity; attacks add overhead but do not dominate hybrid energy cost. Error Packets and Attacks Applied increase from S0→H-S→H-M, mirroring PDR decline and exposing in-path SRH tampering. Roaming amplifies disruption by sweeping the attacker across more cut-sets, increasing on-path exposure, exacerbating mid-flight plan staleness, and exploiting mobility-driven churn that repeatedly refreshes routing opportunities. Practically, mixed-mobility IoT networks are highly exposed: a single roaming attacker can degrade command delivery over wide areas. Ensuring downward path integrity through lightweight SRH checks is essential, while operational monitoring should correlate PDR dips with spikes in Error Packets. Mobility-aware responses—such as quarantining nodes linked to recurring errors, tightening route-plan TTLs, or avoiding recently implicated segments—can help reduce exposure windows in hybrid deployments.

4.0 CONCLUSION

Hatchetman consistently suppresses delivery whenever it sits in-path: in static (S0) networks, PDR reductions directly reflect the number of successful SRH tamper events and resulting errors, while off-path attackers leave metrics unchanged, providing a natural negative control. Mixed mobility further amplifies exposure, as path churn increases the likelihood that legitimate downward routes traverse the attacker, producing sustained PDR loss relative to the hybrid baseline. Roaming attackers cause the greatest disruption by intersecting more distinct routes over time, yielding lower PDR and higher error counts than static attackers under identical density and traffic. Latency, control-plane airtime, and energy remain predictable: AE2ED generally decreases due to survivorship of shorter successful deliveries; DIO traffic keeps airtime clustered around 44–49%; and energy rises primarily due to mobility rather than the attack itself. The study contributes the first comparative evaluation across mobility mixes with explicit static vs. roaming attacker separation; a lightweight, reproducible metric suite; an executable SimPy framework that faithfully models per-packet SRH plans and in-path near-tail tampering with warm-up activation; practical attacker placement for static meshes; and ten-seed batches that capture variability. Future work includes adding full Trickle and DAO/DAO-ACK fidelity, studying storing-mode, testing richer mobility and multi-sink models, exploring probabilistic SRH edits and multi-attacker campaigns at larger scales, evaluating lightweight path-integrity checks and anomaly detection, refining energy and churn measurements, and validating findings on Contiki-NG/RIOT testbeds against classical RPL attacks. Overall, downward SRH manipulation emerges as a practical, stealthy threat in mixed-mobility IoT meshes, with roaming adversaries exerting the strongest impact and motivating mobility-aware monitoring and lightweight path-integrity defenses.

REFERENCES

Abuagoub, M., &Alodat, M. T. (2024). *Hybrid trust-based model for enhancing RPL*

security against internal attacks in IoT networks. IEEE Access, 12, 104857–104872. https://doi.org/10.1109/ACCESS.2024.XXXX

Alfriehat, N., Anbar, M., Aladaileh, M., Hasbullah, I., Shurbaji, T. A., Karuppayah, S., & Almomani, A. (2024). *RPL-based attack detection approaches in IoT networks: Review and taxonomy.*Artificial Intelligence Review, 57, 248. https://doi.org/10.1007/s10462-024-10907-y

Aydin, I., Arslan, M., & Malik, A. S. (2024). *Security vulnerabilities and mitigation techniques in RPL for IoT networks: A comprehensive survey*. Computers & Security, 140, 103628. https://doi.org/10.1016/j.cose.2024.103628

Goel, A., Saxena, N., & Kim, H. (2024). *Machine learning-assisted RPL intrusion detection for resource-constrained IoT*. Future Generation Computer Systems, 153, 312–328. https://doi.org/10.1016/j.future.2024.01.004

Mali, S. D., & Govinda, K. (2021). *A study on network routing attacks in IoT*. Materials Today: Proceedings.https://doi.org/10.1016/j.matpr.2021.07.092

Pu, C., & Song, T. (2018). *Hatchetman Attack: A denial of service attack against routing in low power and lossy networks*. 2018 4th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom) (pp. 12–17). IEEE. https://doi.org/10.1109/CSCloud/EdgeCom.2018.00012

Sejaphala, B. F., Dlamini, S., & Abu-Mahfouz, A. M. (2024). *Lightweight anomaly-based IDS for RPL networks in IoT environments*. Sensors, 24(6), 2229. https://doi.org/10.3390/s24062229

Sharma, G., Grover, J., Verma, A., Kumar, R., &Lahre, R. (2022). Analysis of Hatchetman attack in RPL-based IoT networks. In Proceedings of the International Conference on Computing, Communication, and Intelligent Systems (ICCCIS) (pp. 1–14). IEEE.

Sharma, G., Grover, J., & Verma, A. (2023, December). *A lightweight security solution for mitigation of Hatchetman attack in RPL-based 6LoWPAN*. In 2023 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS) (pp. 1–7). IEEE. https://doi.org/10.1109/ANTS59832.2023.10469481

Stephen, L., Muduli, L., Das, H. S., & Dhiman, G. (2021). Attacks against RPL in IoT: A survey. Materials Today: Proceedings, 46, 11388–11394. https://doi.org/10.1016/j.matpr.2021.02.466

Zilberman, A., Dvir, A., & Stulman, A. (2024). SPRINKLER: A multi-RPL man-in-the-middle identification scheme in IoT networks. IEEE Transactions on Mobile Computing, 23(10), 9971–9985. https://doi.org/10.1109/TMC.2024.3370894