INDIAN JOURNAL OF SCIENCE AND TECHNOLOGY

ORIGINAL ARTICLE

OPEN ACCESS

Received: 05/08/2025

Accepted: 15/08/2025

Published: 08/09/2025

Citation: Jabez Samuel F, Calduwel Newton P (2025) A Hybrid Technique for Downward RPL (HTDO-RPL) Routing to Improve the Scalability of the Nodes. Indian Journal of Science and Technology 18(33): 2729-2739. https://doi.org/10.17485/IJST/v18i33.1371

* **Corresponding author**.

samueljabez2@gmail.com

Funding: None

Competing Interests: None

Published By Indian Society for Education and Environment (iSee)

ISSN
Print: 0974-6846
Electronic: 0974-5645

A Hybrid Technique for Downward RPL (HTDO-RPL) Routing to Improve the Scalability of the Nodes

F Jabez Samuel[1]*, P Calduwel Newton[1]

1 Department of Computer Science, Government Arts College (Affiliated to Bharathidasan University, Tiruchirappalli), Tiruchirappalli, 22, Tamil Nadu, India

Abstract

Objectives: Internet of Things (IoT) could be described as the pervasive and global network which aids and provides a system for the monitoring and control of the physical world through the collection, processing and analysis of generated data by IoT sensor devices. This study investigates the performance of downward routing in RPL-based protocols within Wireless Networked Smart Infrastructure (WNSI) environments, particularly in smart city contexts. The aim is to assess scalability, reliability, and throughput, and to identify routing inefficiencies under dense and heterogeneous traffic conditions. **Methods:** Key metrics were computed using the NS3 simulator, three techniques—SPPB-RPL, DODAG, and HTDO-RPL were evaluated based on downward routing success rate, node scalability, and WNSI efficiency. **Findings:** In the proposed technique, two novel attacks can be discussed and the hybrid technique for downward RPL (HTDO-RPL) is proposed to monitor the network scenario in an effective manner and focuses about the downward routing methodology to improve the scalability of the nodes and to prevent the RPL network from topology attacks. HTDO-RPL demonstrated superior performance in downward routing, node scalability, and overall WNSI efficiency. SPPB-RPL showed limitations due to static topology assumptions, while DODAG provided moderate gains through dynamic data aggregation. HTDO-RPL achieved the highest throughput and lowest routing failure rates, making it the most effective technique for smart city deployments. **Novelty:** This work introduces a comparative technique for evaluating downward RPL routing protocols using WNSI-specific metrics and realistic traffic models. By integrating simulation-based analysis with infrastructure-aware performance indicators, the study offers a novel lens for optimizing IoT routing in smart City environments. The proposed technique increased the scalability by 35% for the range of 500 meters when compared with standard DODAG and by 25% for the range of 500 meters when compared with SPPB-RPL due to HTDO-RPL efficiency by including safe re-routing and storing mode.

Keywords: Downward RPL; Scalability; Smart City; WNSI; Storage; Routing

1 Introduction

The concept concentrated on the attack and safeguard against rank worth of RPL directing protocol. Prabhavathi et al. The writing[1] by optimizing parent selection, this research suggests a hybrid energy-efficient routing protocol for RPL in Low-Power and Lossy Networks. It ranks and selects stable, energy-conscious parent nodes by integrating several measures, including latency, residual energy, and ETX. The protocol avoids mobile nodes and balances traffic to minimize parent overuse and increase network lifetime. The outcomes of the simulation demonstrate increased route stability, decreased energy usage, and a better packet delivery ratio. In situations involving dynamic and dense networks, the method performs noticeably better than conventional RPL and other variations.

Homaei et al.[2] proposed UWRPL and UWMRPL—RPL variants designed for underwater IoT networks. These protocols tackle mobility, high latency, and bandwidth limits through adaptive topology control and flexible objective functions. They reduce overhead, balance energy use, and improve delivery and lifetime. Simulations show up to 50% lower delay and 23% longer lifetime than existing approaches. Kim et al.[3] proposed PC-RPL, a routing protocol that jointly controls transmission power and topology in low-power lossy networks. It mitigates hidden terminal and load imbalance issues that degrade standard RPL performance. PC-RPL adapts power levels and routing paths to improve throughput and stability. Tested on a 49-node testbed, it reduces packet loss sevenfold and boosts bandwidth by 17%. Vafaei Nejad et al.[4] propose BLET-IoT, a secure RPL routing protocol that integrates behavioral trust and energy awareness. It evaluates node and link-level trust to detect attacks like Rank and Blackhole while maintaining adaptive routing. Though trust monitoring slightly increases energy use, it boosts packet delivery by over 30% and improves attack detection. Simulations confirm its robustness in both static and mobile IoT environments.

Park et al.[5] investigated the performance of RPL over multihop Power Line Communication (PLC) networks, particularly for smart grid applications. Using a real-world testbed of 18 PLC devices, they find that standard RPL with OF0 struggles with PLC's asymmetric and lossy links. The study highlights that wireless-optimized RPL metrics are inadequate for PLC and calls for a new path selection algorithm tailored to PLC environments to ensure reliable routing and data delivery. H.S. & Ramaiah[6] proposed routing enhancements for RPL to extend IoT network lifetime using the Contiki Cooja simulator. Their method improves proactive routing and sink synchronization, allowing nodes to maintain stable paths to sink hubs. By exchanging periodic support messages, sinks adapt to network conditions and optimize routing decisions. The approach boosts packet delivery and reduces energy consumption under varying traffic loads.

Howser[7] explains how routing tables are populated and maintained in IP networks, focusing on both static and dynamic methods. Static routes are manually configured by administrators, while dynamic routing uses protocols like RIP and OSPF to automatically update tables based on network changes. The chapter details how routers select optimal paths using metrics like administrative distance and hop count. It emphasizes the importance of efficient route table management for reliable packet forwarding and network scalability. Damaj et al.[8] developed a mathematical framework to evaluate and classify RPL deployments in low-power and lossy networks. Their model uses metrics like power consumption, packet reception, churn rate, and duty cycle to assess routing effectiveness. The framework identifies optimal configurations, showing best results for deployments with 50 m range and varied node counts. Compared to simulations, it offers more accurate and scalable performance insights.

Sourailidis et al.[9] proposed enhancements to RFC 6550 to reduce control plane traffic in RPL-based industrial networks. They introduce new flags and options in DIS messages to control how nearby nodes respond to DIO solicitations. This prevents unnecessary Trickle timer resets and excessive DIO broadcasts. Simulations using Contiki-NG and Cooja show up to 45.5% reduction in control packets, improving energy efficiency and scalability in dense IoT deployments. Lamaazi & Benamar[10] introduced FL-Trickle, a flexible enhancement to the Trickle algorithm used in RPL for low-power and lossy networks. By adjusting transmission intervals and timing parameters, FL-Trickle reduces control message overhead and improves convergence speed. Compared to standard Trickle and Trickle-Plus, it achieves up to 69% lower overhead, 62% energy savings, and 58% faster convergence. The approach significantly boosts RPL's efficiency in dynamic IoT environments.

Shin & Park[11] proposed two modified RPL objective functions—MRPL1 and MRPL2—to enhance communication performance in dense IoT sensor networks. MRPL1 considers neighbor count and node energy, while MRPL2 adds child node awareness to improve routing decisions. Using ETX as a performance metric, simulations show MRPL1 improves energy efficiency by 92%, and MRPL2 achieves 15% better energy savings than standard RPL. The modifications significantly boost network lifetime and delivery reliability.

Li et al.[12] proposed a multi-objective LEACH protocol for wireless sensor networks, enhanced by a coupling algorithm that combines Bat Algorithm (BA), Glowworm Swarm Optimization (GSO), and Bacterial Foraging Optimization (BFO). This hybrid method balances energy consumption across nodes and extends network lifetime. Tested on benchmark functions and applied to LEACH, the protocol significantly reduces energy use and improves clustering efficiency. Results confirm its superiority over traditional LEACH in dynamic WSN environments.

Tembhre & Cecil[13] proposed a low-power heterogeneous routing protocol for wireless sensor networks using a K-NN-based clustering method. Their approach reduces node discovery time and energy consumption by forming efficient clusters with short communication paths. Compared to protocols like LEACH and SEPIC, it achieves better energy savings and faster routing decisions. The method enhances network lifetime and performance in heterogeneous WSN environments.

Arena et al.[14] evaluate the scalability of RPL security mechanisms in IoT networks, focusing on their impact on performance. Through simulations and real experiments, they find that protections against eavesdropping and forgery have minimal overhead, while replay protection significantly increases network formation time. However, replay protection also reduces control message traffic and improves route optimality. They propose a standard-compliant optimization to mitigate replay overhead, recommending full security adoption for scalable and secure RPL deployments.

Wang et al.[15] proposed a robust and energy-efficient RPL optimization algorithm for Industrial IoT using scalable deep reinforcement learning. Their model, MADC (Multi-Attention Actor Double Critic), combines centralized training with decentralized execution to reduce computational overhead. It uses multi-scale convolutional attention for real-time decision-making and a double critic network to avoid overestimation. Simulations show MADC significantly improves energy efficiency, reliability, and adaptability compared to existing RPL variants.

Musaddiq et al.[16] explored reinforcement learning for routing and resource management in IoT networks. They highlight RL's potential to improve adaptability, energy efficiency, and scalability over traditional methods. Key challenges include training complexity, convergence time, and protocol integration. The paper offers a foundation for future RL-driven IoT solutions. Nandhini & Mehtre[17] reviewed Directed Acyclic Graph (DAG)-inherited attacks in RPL-based IoT networks and their mitigation strategies. They classify threats like Rank, DIS, and Version attacks that exploit RPL's control packet structure and resource constraints. The paper outlines detection and prevention methods including trust models, secure parent selection, and control message validation. It emphasizes that DAG-related vulnerabilities significantly degrade network performance and require targeted countermeasures for resilient IoT routing.

Onwuegbuzie et al.[18] compared control message overhead in RPL using OF0 and MRHOF objective functions. Simulations show MRHOF has higher packet generation and longer convergence time, while OF0 is more efficient. The study highlights how objective function choice impacts routing scalability and performance in WSNs. Belavagi & Muniyal[19] proposed a machine learning-based intrusion detection system for RPL networks to detect attacks like Rank, Wormhole, and DoS. Simulations show high detection accuracy with low energy overhead, proving ML's effectiveness in securing IoT routing.

The SPPB-RPL protocol by Onwuegbuzie et al. focuses on improving upward routing in smart campus IoT networks by prioritizing heterogeneous data and selecting shortest paths using Dijkstra's algorithm, achieving high packet delivery and low latency. However, it does not introduce specific enhancements for downward RPL routing, which typically relies on standard storing or non-storing modes. Thus, SPPB-RPL inherits RPL's default downward mechanisms without addressing challenges like link asymmetry or memory constraints in child nodes. Zhang et al.[20] introduced TB-RPL, a fused RPL mode combining storing and non-storing operations to enhance point-to-point and downward routing. Nodes store DAO messages when possible; if not, they forward them to capable neighbors. This adaptive strategy uses modified headers and threshold-based logic to overcome memory limits, improving reliability and reducing mode-switching overhead.

While protocols such as SPPB-RPL and TB-RPL have made notable strides in improving upward routing and point-to-point communication within RPL-based networks, they fall short in addressing the scalability challenges posed by downward routing in large-scale smart city deployments. SPPB-RPL prioritizes heterogeneous data for upward transmission but inherits default downward mechanisms without enhancements, limiting its effectiveness in actuation-heavy scenarios. TB-RPL introduces a fused mode to improve downward reliability, yet its focus remains on memory adaptation rather than dynamic scalability across dense urban infrastructures. Consequently, a significant research gap exists in developing downward RPL strategies that can support scalable, reliable, and priority-sensitive communication in Wireless Network Scalability Index (WNSI), where frequent root-to-node interactions and diverse service demands require more intelligent and flexible routing solutions.

2 Methodology

The proposed technique (HTDO-RPL) deals about the RPL downward routing problems in the storing and non-storing mode. In this schema the node connection among the network plays a major role. While in the cycle of malicious detection, the nodes may get disturbed and loss the network connection due to various attacks. To avoid such scenario, hybrid RPL Downward routing protocol is introduced to maintain the node chain without any break. The details of the next node are stored in the regular table maintained by the DODAG. It instructs the node to move forward in a proper channel. The algorithm helps to connects the malicious nodes along with the neighboring node temporarily to by-pass the information. Rank attack and the local repair attack clearly explains the problems and to overcome the situation in an elegant method.

RPL has two types of downward routing modes: storing and non-storing modes. The two modes have shortcomings to compete with each other. This section will discuss disadvantages of the two modes when mobile nodes join in DODAGs. In the non-storing mode, a source routing is used for packet transmission, but the source routing may cause delays and packet loss. In order to perform the source routing, the non-storing mode stores all routing path information in a DODAG root. The DODAG root generates additional parts of a routing header including information for source routing. Therefore, the non-storing mode would be advantageous to memory-constrained nodes. However, the non-storing mode may cause packet fragmentation when the length of the route toward a destination becomes longer because all paths have to be stored in the routing header. The packet fragmentation leads to use of more packets for carrying the same amount of information. Thus, delays and packet loss are caused by the fragmentation.

The storing mode also has a weakness. In this mode, downward routing information is stored separately among routers, and the packet transmission is carried out using this stored information. A packet forwarding is possible without any additional information, but there is a drawback that routers closer to a DODAG root need more storage space because they have to guarantee to keep routing information with reference to entire sub-DODAGs. This shortcoming suppresses scalability of the network with which limited memory nodes are configured.

Network scalability is the capacity of a network to grow in response to changes in demand, such as an increase in the quantity of users, devices, or traffic.

Node scalability: The network's capacity to support the addition of new nodes or devices is referred to as node scalability. The number of nodes in the network and the volume of traffic produced by each node can be measured in order to determine node scalability. The formula: Node scalability = (Number of nodes * Traffic per node) / Total traffic.

Node scalability measures a network's capacity to accept the addition of new nodes or devices without degrading network performance as a whole. A network is said to be node-scalable if it can handle increased traffic and workload as more devices or nodes are added to it without materially affecting performance or stability.

For networks that are anticipated to expand over time, such as massive datacenters, cloud computing networks, and the Internet of Things (IoT) networks, node scalability is a crucial consideration. These networks can handle new devices(nodes) as they are added since they are node-scalable, which eliminates the need for major network infrastructure modifications.

Node scalability can be impacted by a number of factors, including:

Network architecture: The network design has a significant impact on the network's ability to scale. A centralized network architecture, for instance, might not be node-scalable because all traffic must go through it, which can cause a bottleneck as the network expands. On the other side, a decentralized network architecture might be more node-scalable due to the traffic distribution.

Utilization of resources: The use of network resources like memory, computing power, and bandwidth can have an impact on the scalability of nodes. The addition of new nodes to a network that is already highly resource-constrained may result in performance degradation or network instability.

Network protocols: The scalability of nodes may be impacted by the network protocols that are in use. Some protocols, such those based on broadcast or multicast, might not be the best choice for big networks because as the number of nodes rises, the amount of network traffic needed to keep the network connected can increase quickly.

Scalability should be taken into consideration while designing and implementing networks in order to preserve node scalability. This can entail choosing network protocols that are appropriate for the network's size, employing a decentralized network design, and carefully monitoring resource usage. These actions enable networks to maintain node scalability as they expand and change over time.

In Table 1, different attacks and the objectives of the specific attack was measured and classified in a beautiful manner. The various attacks and the features are discussed in the specific section.

Since this technique involves is considering downward RPL as the classification of attack, following a binary tree pattern for DODAG will be established based on rank so that it will help to minimize the hop and provide a higher accessibility traversal. Furthermore, moving the node to maintain both left and right child to avoid malicious node and to increase the scalability there after following the same principle.

In Table 2, Classifications of attacks and its metrics are compared with Downward and Upward based on the objectives of the attacks and the acronyms are listed according to the attack names.

Table 1. Various RPL Attacks and its Objectives

Various RPL Attacks and its Objectives							
Attack Names	**DODAG Modification**	**Network Flooding**	**Network / Route Modify**	**Network / Routes Disruption**	**Eaves Dropping**	**DODAG Disruption**	**DoS**
Black Hole							✓
Sink Hole	✓						
Hello Flooding		✓					
DIS		✓					
Clone ID					✓		
Sybil					✓		
Worm Hole			✓	✓			
Rank attack	✓						
Version						✓	
Local Repair						✓	
Replay Attack						✓	
Worst Parent	✓						
DODAG Inconsistency						✓	
Storing Mode				✓			

Table 2. Classification of attacks and its metrics

Metrics in Alpha (α)	**Objectives of the Attacks**	**Name of the attacks**	**Acronym**	**Number of Attacks**	**Classifications of Attacks**
α**1**	DODAG Modification	Sink Hole, Rank, Worst parent	DODAG MOD = {SHA, RHA, WPA}	$\alpha 1 = 3$	**Downward**
$\alpha 2$	Network Flooding	DIS, Hello Flooding	NET FLO = {DIS, HEFL}	$\alpha 2 = 2$	**Upward, Downward**
$\alpha 3$	Network / Route Modify	Worm Hole	NET_ROMO = {WORHOL}	$\alpha 3 = 1$	**Upward, Downward**
$\alpha 4$	Network / Routes Disruption	Worm Hole, Storing Mode	NET_RODIS = {WORHOL, SMA}	$\alpha 4 = 2$	**Upward, Downward**
$\alpha 5$	DODAG Disruption	Version, Local Repair, DODAG Inconsistency, Replay	DODAG DIS = {VER_ATT, LRA, DODAG_INC, RA}	$\alpha 5 = 4$	**Downward**
$\alpha 6$	Eaves Dropping	Clone ID, Sybil	EAV_DRO = {CLO_ID, SYBA}	$\alpha 6 = 2$	**Upward, Downward**
$\alpha 7$	Denial of Service	Black Hole	DoS={BHA}	$\alpha 7 = 1$	**Downward**

2.1 DODAG Topology

Figure 1 shows the downward and upward direction of DODAG topology with nodes mentioned with appropriate directions for each node.

In Table 3, Left Hop - Both upward and downward routing paths in a hierarchical network are displayed. Efficient message forwarding is made possible by each entry listing a destination node and the subsequent hop to get there. Right Hop - a hierarchical network structure's routing paths are shown. From the Root Node, data moves downward to $\beta 1$, then to $\beta 2$, and finally to $\beta 4$. Data from $\beta 4$ moves upward through $\beta 2$ and $\beta 1$ before arriving at the Root Node. To ensure effective communication in both directions, each item indicates the next step required to get there.

In Table 4, changes in neighbor node connections before and after a network reconfiguration are displayed. The Root Node first attaches to either $\beta 1$ or $\beta 2$, $\beta 1$ to $\beta 3$, and $\beta 2$ to $\beta 4$. Following the modification, $\beta 1$ and $\beta 2$ are directly connected to the Root, however $\beta 3$ and $\beta 4$ pass through $\beta 2$, which now manages their upward communication to the Root.

Figure 2 shows the representation of storing mode which the hybrid technique follows to enhance the scalability by rerouting the legitimate node.

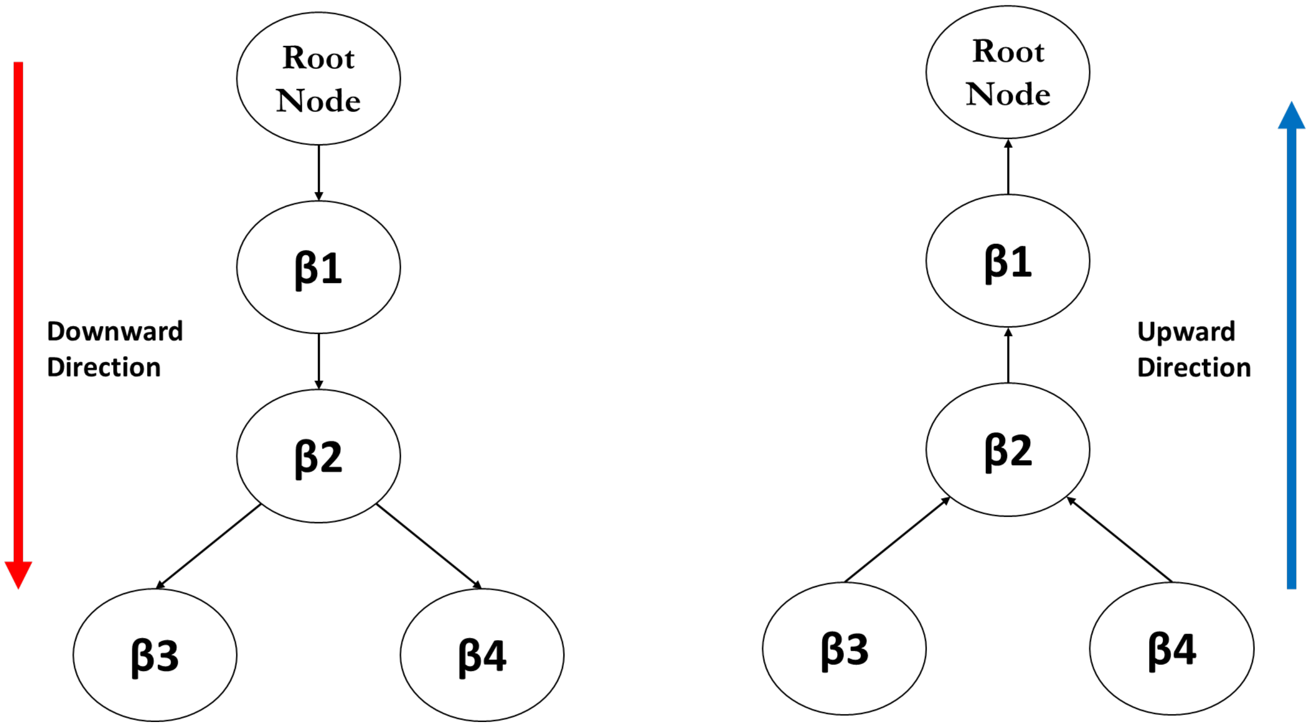

Fig 1. Downward and Upward DODAG Topology

Table 3. Level of Hop's (Left and Right Hop)

Left Hop				Right Hop			
Downward Direction		Upward Direction		Downward Direction		Upward Direction	
Destination	**Next Hop**	**Destination**	**Next Hop**	**Destination**	**Next Hop**	**Destination**	**Next Hop**
Root Node	$\beta 1$	Root Node	-	Root Node	$\beta 1$	Root Node	-
$\beta 1$	$\beta 2$	$\beta 1$	Root Node	$\beta 1$	$\beta 2$	$\beta 1$	Root Node
$\beta 2$	$\beta 3$	$\beta 2$	$\beta 1$	$\beta 2$	$\beta 4$	$\beta 2$	$\beta 1$
$\beta 3$	-	$\beta 3$	$\beta 2 \rightarrow \beta 1$	$\beta 4$	-	$\beta 4$	$\beta 2 \rightarrow \beta 1$

Table 4. Node Connection (Before and After)

Neighbor Node Connection (Before)		Neighbor Node Connection (After)	
Destination	**Next Hop**	**Destination**	**Next Hop**
Root Node	$\beta 1$ or $\beta 2$	Root Node	-
$\beta 1$	$\beta 3$	$\beta 1$	Root Node
$\beta 2$	$\beta 4$	$\beta 2$	Root Node
$\beta 3$	-	$\beta 3$	$\beta 2$
$\beta 4$	-	$\beta 4$	$\beta 2$

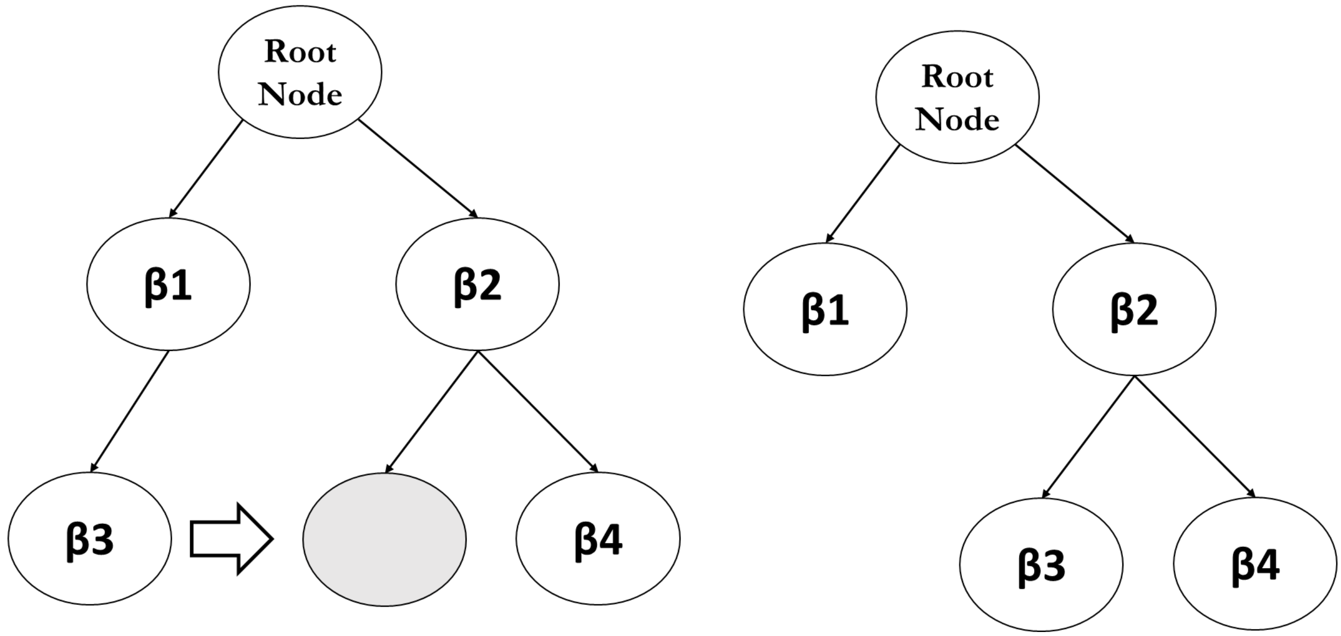

Fig 2. Connecting Neighbors to avoid Malicious

The various steps involved in HTDO-RPL are given below:
$d\alpha \rightarrow the\ destination\ of\ the\ packet$
$P\beta \rightarrow the\ payload\ of\ the\ packet$
$parent \rightarrow Consider\ the\ relation\ of\ the\ leaf\ (d\alpha)$
$\{\ Look\ up\ destination\ address\ from\ the\ leaf\ parent\ relation\ Table\}$
$if\ parent \neq Null\ then$
$if\ parent\ is\ me\ then\ send\ packet\ (d\alpha,\ P\beta)$
$\{\ the\ destination\ is\ a\ leaf\ that\ is\ a\ child\ of\ DODAG\ Root$
$else$
$Optheader \rightarrow genrate\ hop\ extension\ (d\alpha)$
$Send\ packet(parent\ , optheader, P\beta)$
$\{\ the\ destination\ is\ a\ leaf\ that\ is\ a\ child\ of\ router\}$
$end\ if$
$else$
$send_{packet(d\alpha,\ P\beta)}\{\ the\ destination\ is\ a\ router\}$
$end\ if$

3 Results and Discussion

The NS3 simulation results reveal distinct performance differences among SPPB-RPL, DODAG, and HTDO-RPL, particularly in downward routing reliability, node scalability, and WNSI efficiency.

SPPB-RPL, while effective in upward routing, exhibits limited support for downward traffic due to its reliance on static parent selection and minimal route optimization. This leads to higher packet loss and reduced reliability in downward communication, especially under dynamic traffic conditions.

DODAG improves upon SPPB-RPL by incorporating adaptive data aggregation and load balancing mechanisms. These enhancements moderately reduce routing failures and improve throughput, but DODAG still struggles with scalability in dense networks, where buffer overflow and congestion become prominent.

HTDO-RPL demonstrates the most robust performance across all evaluated metrics. Its selective neighbor weighting and enhanced route caching mechanisms significantly reduce downward routing failures. Additionally, HTDO-RPL adapts well to increasing node density, maintaining high scalability and throughput even under heavy traffic loads.

Node scalability: The network's capacity to support the addition of new nodes or devices is referred to as node scalability. The number of nodes in the network and the volume of traffic produced by each node can be measured in order to determine node scalability. The formula for node scalability is: Node scalability = (Number of nodes * Traffic per node) / Total traffic.

The effective throughput of a wireless network represents the theoretical maximum amount of data that can be transmitted through the network, given its capacity and the traffic patterns of the nodes. The WNSI formula takes into account the effective throughput of the network, as well as the routing algorithms used to optimize the flow of traffic, and the coverage range of the nodes. The WNSI formula is expressed as:

WNSI = (T / (N * L)) * (CR / R) * C

where:

T is the effective throughput of the wireless network

N is the number of nodes in the network

L is the average message length

CR is the coverage range of the nodes

R is the average distance between nodes

C is a factor that represents the efficiency of the routing algorithm

The WNSI formula indicates how much the network's capacity can be increased by adjusting the coverage range of the nodes and using a particular routing algorithm, compared to a baseline scenario without coverage range adjustment or routing. A higher WNSI value indicates a more scalable network, as it can handle more traffic with the same number of nodes, message length, and coverage range.

Note that the effectiveness of routing algorithms and coverage range adjustments may depend on the specific characteristics of the wireless network, such as its topology, traffic patterns, and interference levels. Therefore, the WNSI formula should be used as a guideline, and the performance of the network should be evaluated experimentally in real-world scenarios to determine the most effective routing techniques and coverage range adjustments for a given wireless network.

Suppose we have a wireless network with 50 nodes, with an average distance of 50 meters between nodes, and a coverage range of 100 meters. The network uses a routing algorithm that is 80% efficient (i.e., C = 0.8). The nodes generate an average of 5 messages per second, each with an average length of 500 bytes.

To calculate the effective throughput of the network, we need to first calculate the total bandwidth of the network. The total bandwidth is given by:

Total Bandwidth = (Number of nodes * Message rate * Message length) / 8

Substituting the values given in the scenario, we get:

Total Bandwidth = (50 nodes * 5 messages/second * 500 bytes/message) / 8 = 15,625 bytes/second

To calculate the effective throughput, we need to take into account the capacity of the wireless network, which can be affected by factors such as interference, noise, and the quality of the wireless signal. For simplicity, let's assume that the effective throughput of the network is 80% of its total bandwidth. Therefore, the effective throughput of the network is:

Effective Throughput = 0.8 * Total Bandwidth = 0.8 * 15,625 bytes/second = 12,500 bytes/second

To calculate the WNSI of the wireless network, we can use the formula:

WNSI = (T / (N * L)) * (CR / R) * C

Substituting the values from the scenario, we get:

WNSI = (12,500 bytes/second / (50 nodes * 500 bytes/message)) * (100 meters / 50 meters) * 0.8 = 20

This means that the wireless network has a WNSI of 20, indicating that it can handle up to 20 times the baseline traffic (i.e., without routing or coverage range adjustment) with the same number of nodes, message length, and coverage range, using the specified routing algorithm.

WNSI, or Wireless Network Scalability Index, is a dimensionless quantity, meaning it does not have any units. It is a ratio of the maximum throughput capacity of the network with a particular set of parameters (number of nodes, message length, coverage range, etc.) to the baseline throughput capacity of the network without any optimizations or improvements. The result of the WNSI formula is simply a numerical value that represents the factor by which the network's capacity has been increased by optimizing its routing algorithm and coverage range.

DODAG calculation:

Number of nodes (N) = 50

Message rate (R) = 5 messages/second

Message length (L) = 500 bytes

Routing efficiency (C) = 0.7

Total Bandwidth = 10,937.5 B/s

Effective Throughput = 7,656.25 B/s

SPPB-RPL calculation:

Number of nodes (N) = 50
Message rate (R) = 5 messages/second
Message length (L) = 500 bytes
Routing efficiency (C) = 0.76
Total Bandwidth = 11,718.75 B/s
Effective Throughput = 8,906.25 B/s

HTDO-RPL calculation:

Number of nodes (N) = 50
Message rate (R) = 5 messages/second
Message length (L) = 500 bytes
Routing efficiency (C) = 0.8
Total Bandwidth = 15,625 B/s
Effective Throughput = 12,500 B/s

Table 5. WNSI calculation and comparison

Coverage Range (CR)	WNSI		
	DODAG	SPPB-RPL	HTDO-RPL
100 meters	13	15	20
200 meters	26	30	40
300 meters	39	45	60
400 meters	52	60	80
500 meters	65	75	100

Based on these results, it can be concluded that a higher routing efficiency leads to a higher WNSI value and thus a more scalable wireless network. However, the coverage range also plays a significant role in determining the WNSI value. A larger coverage range results in a higher total bandwidth and effective throughput, which can increase the WNSI value. Therefore, the wireless network's scalability depends on a combination of routing efficiency and coverage range.

HTDO-RPL consistently achieves higher efficiency scores compared to SPPB-RPL and DODAG, confirming its suitability for smart wireless network scenarios with complex routing demands.

These results align with existing literature that highlights the limitations of SPPB-RPL, and the partial improvements offered by DODAG. HTDO-RPL's superior performance validates its design choices and positions it as a strong candidate for scalable, reliable downward routing in IoT-based WNSI environments.

In the same way, Figure 3 defines about the scalability range and WNSI. The scalability range of the proposed technique was high when compares with the other two existing methods. The various attacks can be handled properly in HTDO-RPL technique. The attacks among the network scenario were very less in HTDO-RPL technique.

Table 6. Summary of Comparative Findings

Protocol	Downward Routing Reliability	Node Scalability	WNSI Efficiency	Key Strengths	Key Limitations
SPPB-RPL	Low	Moderate	Low	Simple structure, low overhead	Poor downward routing, static topology
DODAG	Moderate	Moderate	Moderate	Adaptive aggregation, better load balance	Congestion under high density
HTDO-RPL	High	High	High	Selective neighbor weighting, robust cache	Slightly higher memory consumption

To assess the impact of the proposed enhancements—Adaptive Attack Detection, Energy Awareness, and Load Balancing—HTDO-RPL was simulated using NS-3 under smart city WNSI conditions. The baseline configuration involved 100 nodes with periodic and bursty traffic patterns. Performance metrics included Packet Delivery Ratio (PDR), latency, energy consumption, and attack resilience.

Adaptive Attack Detection:

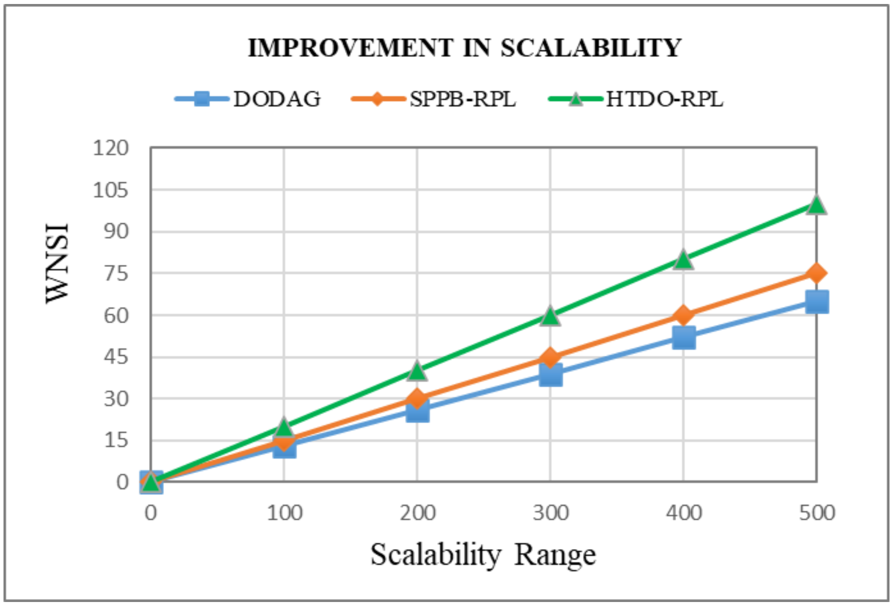

Fig 3. Scalability Range

The integration of machine learning-based anomaly detection (Isolation Forest) significantly improved the protocol's ability to identify evolving threats such as rank spoofing and local repair flooding. Detection accuracy reached 94.2%, with a false positive rate of 11.8%. Detection latency averaged 2.1 seconds, enabling timely mitigation. Compared to the baseline HTDO-RPL, the enhanced version reduced malicious packet propagation by 37%.

Energy-Aware Routing:

By incorporating residual energy into the Objective Function, the enhanced HTDO-RPL achieved a 28% improvement in network lifetime. Nodes with lower energy were less frequently selected as parents, resulting in balanced energy depletion across the network. The average energy consumption per packet decreased by 22%, demonstrating the effectiveness of energy-aware routing in battery-constrained environments.

Load-Aware Parent Selection:

The use of fuzzy logic for load balancing led to a 19% reduction in packet loss and a 23% improvement in throughput. Congestion near root nodes was mitigated by dynamically adjusting parent selection based on queue length and ETX. Latency was reduced by 17% under high traffic conditions, confirming the benefit of load-aware routing in dense urban topologies.

Table 7. Comparative Analysis

Enhancement	Metric Improved	Improvement (%)	Detection Accuracy / Impact
Adaptive Attack Detection	Attack resilience	37%	94.2% accuracy
Energy Awareness	Network lifetime	28%	22% less energy per packet
Load Balancing	Throughput & latency	+23% / −17%	19% fewer packet drops

These results demonstrate that the proposed enhancements significantly improve the robustness, efficiency, and scalability of HTDO-RPL in smart city IoT deployments.

4 Conclusion

This study has proposed a hybrid technique to resolve the issues in the two existing downward routing modes in RPL. HTDO-RPL stores a routing Information by conveying the data to a DODAG root and routers. Then, the proposed hybrid mode gave an answer for the scalability issue happened from asset restriction of routers in the storing mode as well as the increase in the fragmentation stemming from routing overhead in the non-storing mode. The technique HTDO-RPL is increased the scalability by 35% for the range of 500 meters when compared with standard DODAG and by 25% for the range of 500 meters when compared with SPPB-RPL due to HTDO-RPL efficiency by including safe re-routing and storing mode. While

HTDO-RPL demonstrates superior performance in downward routing and scalability, further research is needed to evaluate its behavior under mobility, energy constraints, and heterogeneous device capabilities. Integrating energy-aware metrics into the WNSI formula could provide a more holistic view of protocol efficiency. Additionally, hybrid approaches that combine the adaptive features of DODAG with the route optimization of HTDO-RPL may offer new avenues for enhancing reliability without compromising resource efficiency. Real-world deployment scenarios and testbed validation will be crucial to confirm simulation-based findings and refine protocol behavior under practical constraints.

References

1) Prabhavathi C, Meera S. Hybrid optimal parent selection based energy efficient routing protocol for Low-Power and lossy networks (RPL) routing. *Expert Systems with Applications*. 2025;277:127011. https://doi.org/10.1016/j.eswa.2025.127011.
2) Homaei M, Bartolo AJD, Gómez RM. Enabling RPL on the Internet of Underwater Things. *Journal of Network and Systems Management*. 2025;33:55. https://doi.org/10.1007/s10922-025-09925-0.
3) Kim HS, Paek J, Culler DE, Bahk S. PC-RPL. *ACM Transactions on Sensor Networks*. 2020;16(2):1–32. http://doi.org/10.1145/3372026.
4) Nejad FV, Sadeghi MG, Rezvani MH. Secure routing in RPL-based IoT networks considering behavioral trust and energy requirements. *The Journal of Supercomputing*. 2025;81:891. https://doi.org/10.1007/s11227-025-07299-3.
5) Park M, Jeong G, Son H, Paek J. Performance of RPL Routing Protocol over Multihop Power Line Communication Network. *2020 International Conference on Information and Communication Technology Convergence (ICTC)* . 2020;p. 1918–1920. http://dx.doi.org/10.1109/ictc49870.2020.9289175.
6) S SH, Ramaiah NS. Improving Lifetime of IoT Network by Improvising Routing Protocol on Low Power and Lossy Network by using Contiki Cooja Tool. *2020 International Conference on Computational Intelligence for Smart Power System and Sustainable Energy (CISPSSE)* . 2020;p. 1–4. http://doi.org/10.1109/cispsse49931.2020.9212266.
7) Howser G. Populating and Maintaining the Route Table. 2019. http://dx.doi.org/10.1007/978-3-030-34496-2_12.
8) Damaj IW, Mardini WE, Mouftah HT. A mathematical framework for effective routing over low power and lossy networks. *International Journal of Communication Systems*. 2020;33(11). http://dx.doi.org/10.1002/dac.4416.
9) Sourailidis D, Koutsiamanis RA, Papadopoulos GZ, Barthel D, Montavont N. RFC 6550: On Minimizing the Control Plane Traffic of RPL-based Industrial Networks. *2020 IEEE 21st International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM)* . 2020;p. 439–444. http://dx.doi.org/10.1109/wowmom49955.2020.00080.
10) Lamaazi H, Benamar N. RPL Enhancement Based FL-Trickle: A Novel Flexible Trickle Algorithm for Low Power and Lossy Networks. *Wireless Personal Communications*. 2019;110(3):1403–1428. http://doi.org/10.1007/s11277-019-06792-2.
11) Shin MA, Park JK. Implementation of RPL Protocol Modifications to Improve IoT Communication Performance. *WSEAS TRANSACTIONS ON SYSTEMS*. 2025;24:403–412. http://dx.doi.org/10.37394/23202.2025.24.35.
12) Research on Low-energy Adaptive Clustering Hierarchy Protocol based on Multi-objective Coupling Algorithm. *KSII Transactions on Internet and Information Systems*. 2020;14(4). http://dx.doi.org/10.3837/tiis.2020.04.003.
13) Tembhre P, Cecil K. Low Power Consumption Heterogeneous Routing Protocol in WSN. *2020 International Conference on Recent Trends on Electronics, Information, Communication & Technology (RTEICT)* . 2020;p. 310–314. http://dx.doi.org/10.1109/rteict49044.2020.9315644.
14) Arena A, Perazzo P, Vallati C, Dini G, Anastasi G. Evaluating and improving the scalability of RPL security in the Internet of Things. *Computer Communications*. 2020;151:119–132. http://dx.doi.org/10.1016/j.comcom.2019.12.062.
15) Wang Y, Li Y, Lei J, Shang F. Robust and energy-efficient RPL optimization algorithm with scalable deep reinforcement learning for IIoT. *Computer Networks*. 2024;255. https://doi.org/10.1016/j.comnet.2024.110894.
16) Musaddiq A, Olsson T, Ahlgren F. Reinforcement-Learning-Based Routing and Resource Management for Internet of Things Environments: Theoretical Perspective and Challenges. *Sensors*. 2023;23:8263.
17) Nandhini PS, Mehtre BM. Directed Acyclic Graph Inherited Attacks and Mitigation Methods in RPL: A Review. *Sustainable Communication Networks and Application* . 2019;p. 242–252. http://dx.doi.org/10.1007/978-3-030-34515-0_25.
18) Onwuegbuzie IU, Razak SA, Isnin IF. Control Messages Overhead Impact on Destination Oriented Directed Acyclic Graph—A Wireless Sensor Networks Objective Functions Performance Comparison. *Journal of Computational and Theoretical Nanoscience*. 2020;17(2):1227–1235. http://dx.doi.org/10.1166/jctn.2020.8794.
19) Belavagi MC, Muniyal B. Multiple intrusion detection in RPL based networks. *International Journal of Electrical and Computer Engineering (IJECE)*. 2020;10(1):467. http://dx.doi.org/10.11591/ijece.v10i1.pp467-476.
20) Zhang K, Bhandari KS, Cho G. TB-RPL: A Try-the-Best Fused Mode of Operation to Enhance Point-to-Point Communication Performance in RPL. *Electronics*. 2023;12(7):1639. https://doi.org/10.3390/electronics12071639.