ISSN 2278-3091
Volume 9 No.2, March -April 2020
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse29922020.pdf
https://doi.org/10.30534/ijatcse/2020/29922020

# Towards Effective Multisink Support in IPv6-based IoT Networks

**Mohammed Alreshoodi**
Unizah Community College, Qassim University, Unizah, KSA, mo.alreshoodi@qu.edu.sa

## ABSTRACT

Realizing efficient network routing and data communication is critical for Internet of Things (IoT) systems. Towards this end, the IPv6 Routing Protocol for Low Power and Lossy Network (RPL) has been developed and standardized to provide an effective routing protocol for IoT networks. RPL offers basic and simple network routing that can support a wide range of IoT applications with varying requirements. However, the potentiality of prolonging the RPL model to an advance limit has opened the doors for abundant contributions towards the improvement of RPL functionality. These include the support of topology establishment using more than one sink node and the ability to maintain multisink connectivity. The proposed work in this paper provides a multisink routing solution for prefix-based and hierarchical network topology establishment with aggregated inter-sink routing. The proposed solution was implemented and evaluated considering different simulated scenarios in the Cooja Simulator. The results indicated better network performance in terms of less hop count and better PDR, in addition to maintaining acceptable end-to-end delay.

**Key words :** Wireless Sensor Networks, Low Power and Lossy Networks, Network Routing, Internet of Things.

## 1. INTRODUCTION

Along with the increasing interest in Internet of Things (IoT) applications, there has been a need for a networking model of low deployment cost and less computation complexity. Therefore, Low-Power and Lossy Networks (LLNs) have been utilized to provide effective network infrastructures of energy-efficient and low-power connectivity. However, typical LLN topologies are made of small-sized and resource-limited network devices interconnected over unreliable wireless links of low bandwidth and high rate of loss. Efficient network routing in such a scarce environment is challenging. This would be more difficult for those applications, especially in healthcare and industrial control, requiring reliable and low-energy real-time communications.

Adopting the traditional IP routing protocols to address such a need would be inappropriate, as being of no energy efficiency and network reliability considerations. Therefore, the Internet Engineering Task Force (IETF) provides the IPv6 Routing Protocol for Low Power and Lossy Network (RPL), a customized LLN routing protocol. RPL enables effective IPv6 network routing and traffic control over structured network topology. The design of the RPL protocol is optimized for better energy efficiency and network management. RPL provides a simple routing solution for allowing end-to-end IPv6 architectures across the Internet for more effective IoT applications.

The standard RPL enables the construction of an IoT network with a set of sensor nodes interconnected towards a single sink node in a multihop networking model. The sensor nodes participate and cooperate in forwarding their data packets at each level of the DODAG towards the sink node. Considering multihop routing in a large network setup, most of the data packets would traverse a high number of nodes. This would result in high packet forwarding across the network. Subsequently, the packet loss rate would increase and energy consumption would become higher, leading to a reduction in network lifetime. Since the DODAG is rooted at a single node, traffic load would be saturated at the top-level nodes, increasing the burden on the DODAG sink and adjacent nodes. The DODAG becomes more prone to congested links and overloaded buffers as moving up to the DODAG root. This would also form a networking model of a single point of failure.

One potential approach to alleviate this limitation is extending the DODAG topology with multisink connectivity. The original RPL can be flexibly extended to incorporate multiple sink nodes over which data traffic load can be effectively distributed. Although single-sink connectivity would reduce the protocol complexity and implementation cost, Multisink RPL solutions would improve network performance by different ways such as distributing traffic load, providing better energy balancing, and supporting effective fault tolerance.

This research study is a contribution towards effective multisink support for the standard RPL protocol. The proposed research work in this paper is a multisink solution enabling prefix-based and hierarchical network topology establishment with aggregated inter-sink routing. The proposed solution was implemented and evaluated considering different simulated scenarios in the Cooja Simulator. The results indicated better network performance in terms of less hop count and better PDR, in addition to maintaining acceptable end-to-end delay.

The following section, Section 2, of this research paper provides an overview of the standard RPL protocol and explains its operational functionality. Section 3 presents and discusses the related work. In Section 4, the proposed multisink RPL solution is presented. Section 5 then illustrates the experimental setup for the implementation and evaluation of the proposed solution. Section 6 presents and discusses the evaluation results. The conclusion is presented in Section 7.

## 2. THE RPL PROTOCOL

Low-Power and Lossy Networks (LLNs) refer to wireless networks operating with constrained wireless links and resource-limited network devices. Networks of these kinds are formed with a number of memory, storage, CPU, and energy-limited devices. LLNs communications are established over low bandwidth, high loss rate, and instable wireless links.

At the Physical and MAC layers, LLNs typically use the IEEE 802.15.4 protocol. It guarantees low complexity, power consumption, and cost, with short-range wireless communication and limited data transmission rate. On top of the IEEE 802.15.4, the IPv6 over Low Power Wireless Personal Area Networks (6LowPAN) provides integration solution between LLNs an IPv6 networks. It enables IPv6 communications over LLNs links using an adaptation layer with header compression and fragmentation mechanisms as specified in RFC 4944 [1] and RFC 6282 [2]. At the Network layer, the IPv6 Routing Protocol for LLN (RPL) [3] provides an effective network routing solution for the restricted LLNs environments.

### 2.1 Protocol Overview

RPL provides a distance vector routing solution running at the Network layer. It supports point-to-point, point-to-multipoint, and multipoint-to-point communication schemes. RPL offers a routing framework that enables the implementation of diverse routing optimization objectives with different routing metrics. Therefore, RPL can support wide scope of applications of varying requirements.

RPL constructs a LLN network as a collection of RPL instances; each is formed as a Directed Acyclic Graph (DAG). The structure of an RPL instance consists of one or more Destinations-Oriented DAGs (DODAGs). Figure 1 is an illustration of a case of an RPL network with one instance having two DODAGs. Typically, each DODAG is formed into a multihop structure of multiple RPL nodes rooted at a sink node. The organization of the nodes in the DODAG is boosted in keeping with a certain routing Objective Function (OF). RPL uses OF to construct DODAGs in a way that guarantees optimal network routing. This goes through two main processes, node ranking and parent selection. Each node calculates a rank for itself using the implemented OF. The rank specifies each node's virtual distance to the sink node and inhibits the creation of routing loops. Optimal parent selection can guarantee minimum-cost routing over lowest-ranked parents.

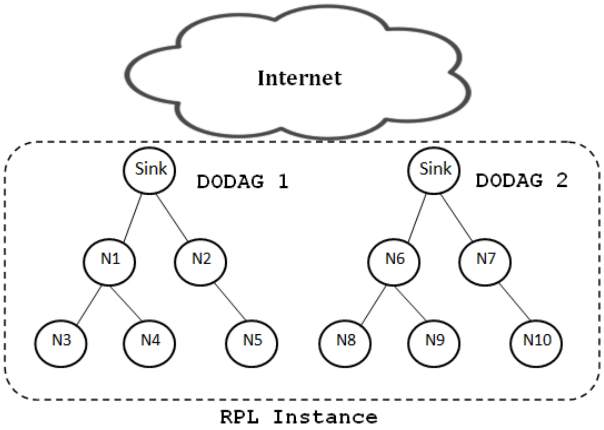

**Figure 1:** RPL Network Architecture.

RPL uses the OF to minimize the cost of reaching the sink and achieve a particular optimization purpose. An OF can be implemented with one or more routing metrics. RFC 6551 [4] defines a number of routing metrics and constraints pertinent to diverse RPL applications. However, there are two default OFs defined for RPL. One is OF0 (Objective Function Zero), presented in RFC 6552 [5]. The other is MRHOF (Minimum Rank with Hysteresis Objective Function), introduced in RFC 6719 [6]. OF0 merely applies the hop count to the sink node as a default routing metric. MRHOF applies Estimated Transmission Count (ETX), a link metric used to estimate the amount of transmissions required for successful packet delivery.

### 2.2 RPL Operations

Each node maintains a Parent List containing all the senders of the DIO messages being received. For attaching to the DODAG, the node applies the advertised objective function on its Parent List. This allows computing its rank value and selecting an optimal parent to attach to. After that, the received DIO is updated and then disseminated further down the DODAG. This enables establishing upward routes in the direction of the DODAG's root and completely building the DODAG. On the other hand, the establishment of the downwards routes requires each RPL node to transmit a Destination Advertisement Object (DAO) message using its upward route. This allows the dissemination of the RPL routing information up to the DODAG's root. Moreover, RPL also defines the DODAG Information Solicitation (DIS) message for eliciting DIO transmission. This is helpful in a situation when a node joining a DODAG receives no DIO broadcasting.

RPL supports two modes for routing downward data traffic. The first is the non-storing mode which is based on source routing and requires all the traffic to be routed through the root. The other mode is the storing mode which is fully stateful. Data traffic in this mode can be routed through a common ancestor in addition to the root. Therefore, DAO messages are unicasted by each node to its preferred parent in storing mode whereas they are unicasted to the root in non-storing mode.

For more effective management of protocol overhead and network stability, the RPL protocol uses the Trickle algorithm as specified in RFC 6206 [7]. The algorithm helps in minimizing control traffic when there are no topological changes in the network. It regulates the DIO transmission interval to exponentially upsurge.

In the cases of node or link failure, two main mechanisms, global and local repair, are supported by RPL to rectify the situation. Global repair is based on rebuilding the DODAG by the sink node, which could be a costly approach considering large deployments. On the other hand, local repair enables simple failure recovery allowing switching the current preferred parent. This is accomplished based on an exchange of DIS and DIO control messages and a reset of the DIO trickle timer. This approach would guarantee successful failover mechanism after increasing the amount of control traffic across the RPL network and incurring additional operational delay.

## 3. RELATED WORK

The RPL design is based on a flexible network routing model allowing for further enhancement of its effectiveness to the maximum. Accordingly, the network research community has been making diverse contributions towards extending RPL functionality. During the recent years, several research works that integrate the support of advance mechanisms into RPL have been proposed. The support of multisink connectivity is one of these mechanisms that extend the RPL network topology to have multiple RPL sink nodes. It is evident that incorporating a multisink approach would improve network performance and durability. It would also facilitate balancing traffic load [8] and maximizing energy efficiency [9-12]. Multisink topologies would also enable addressing efficient QoS solutions [13-14]. The multisink approach can also support effective data collection and aggregation in different WSN applications [15]. A variety of multisink solutions for RPL networks have been introduced by many research works. These can be classified according to their major considerations as presented in Table 1.

As it can be noticed, two main schemes were followed for the support of multisink in RPL. The first is based on enabling a

**Table 1:** Summary of the Reviewed RPL Multisink Solutions

| Solution | Ref. | Multisink Scheme | Support | Evaluation Methodology | Evaluation Measures |
|---|---|---|---|---|---|
| Multisink RPL Performance Study | [16] | Multisink DODAG | Specific application (Smart Home) | Simulation (Cooja) | PDR, Hop Count Rate, ETX Power Consumption, Overhead |
| | [17] | Multisink DODAG | Specific application (SDN-enabled IoT) | Simulation (Cooja) | PDR, Stability, Power Consumption |
| | [18] | Multisink DODAG | Specific application (Smart City) | Simulation (Cooja) | PDR, Node Participation, Power Consumption, Overhead |
| Sink Selection | [19]<br>[20] | Multisink DODAG | Multi-Metric Tiebreaking Approach | Simulation (Cooja) | PDR, Delay, Packet Loss, Retransmission Rate |
| | [21] | Multisink Instance | Failure Recovery Support in Data Collection Systems | Simulation (NS-2) | PDR, Delay, Throughput |
| Centralized Network Architecture | [22] | Multisink DODAG | Virtualization using Anycast Addressing | Simulation (Cooja) & Physical Testbed | Packet Loss, Hop Count Rate, Power Consumption |
| | [23] | Multisink DODAG | Virtualization using a Tunneling Approach & Failure Recovery | Physical Testbed | Delay |
| | [24] | Multisink DODAG and Instance | Physical External Sink & Effective Integration with IPv6 Internet | Physical Testbed | Delay |
| Inter-Sink Coordination | [25]<br>[26] | Multisink Instance | Multi-DODAG Connectivity for RPL Nodes | Simulation (Cooja) | Delay, Throughput, Packet Loss, Network Lifetime |
| Mobility Support | [27] | Multisink DODAG | RPL Node Mobility | Simulation (Cooja) | PDR, Number of Transmission & Re-transmission, Packet Loss, Delay, Overhead |
| | [28] | Multisink Instance | Sink Node Mobility | Simulation (Cooja) & Physical Testbed | Established a Baseline for Timing Configurations |

single DODAG to accommodate more than one sink node and allowing them to operate collectively. This was considered by the researchers in [16-21]. Actually, the specification of the RPL protocol [3] supports a virtualization approach in which multiple sink nodes can coexist in a single RPL DODAG and be coordinated with a virtual DODAG root. This functionality was approached using an anycast addressing approach in [18] and a tunneling-based approach in [19], whereas a different approach was taken in [20] to connect the sink nodes of a DODAG to an external physical root. However, each of these proposals introduces an additional entity to the RPL network model. Although this can facilitate failure recovery [19] and support the integration with the IPv6 network [20], the registration and information exchange among the sink nodes and those entities would incur additional delay and communication overhead. It can also be noticed that data communication is routed over a centralized architecture in these solutions. Data packets follow longer routing paths as being routed all the way to the central entity via different sink nodes in the case of external and internal communication. This would not only cause traffic to traverse sub-optimal paths and increase end-to-end delay, but also add to the protocol complexity and increase control overhead. This is more evident considering the tunneling approach in [19] where every packet needs to be tunneled to reach the central entity. In addition, optimized synchronization among the sink nodes in such a network model is another critical problem that requires effective communications framework.

The other multisink support scheme is based on enabling the coordination among multiple single-root DODAGs of different sink nodes within one RPL instance. This is also adopted by the research works in [20], in addition to [22-25]. Although it is specified that a node in the original RPL can connect to multiple DODAGs in different instances, the proposals in [22-23] are based on breaking this rule as RPL nodes need to listen to DIO messages of other DODAGs in the same instance. This was adopted to facilitate the recovery of sink failure [22] and enable the exchange of sink information among sink nodes [23]. Although this approach could add to network overhead and processing burdens at the edge nodes, it could facilitate inter-DODAG communication with more flexibility and direct interactivity.

It is also evident that the proposed multisink support for RPL took different research directions. One is sink selection considering the availability of different sink nodes. However, it can be seen that such a critical task in multisink setups received less attention in the reviewed literature. It was only recognized in [16-17] following a multi-metrics tiebreaking approach, and in [22] considering basic single-metric sink discovery and selection. Establishing an optimal sink selection framework would be a viable contribution for improving RPL multisink support. Other proposals focused on benefiting from the multisink model to address failure recovery mechanisms [19], [22], but only considering the failure of the sink nodes. Failure recovery at the level of sensor nodes, in particular bottleneck nodes, is still to be addressed.

On the other hand, the need for addressing inter-sink coordination among multiple sink nodes was also considered in [23-24], which enables topology adaptation to current network situations by distributing RPL nodes among different sink nodes. Another consideration is the development of a multisink model with centralized communication and control architecture. Despite the distributed nature of the multisink networking model, this approach was considered for enabling centralized intra-communication [18], failure recovery [19], and integration with IPv6 networks [20]. Other proposals targeted enhancing the mobility support in RPL networks using multisink topologies. This was addressed for mobile RPL nodes in [21] and mobile sink nodes in [25], while considering mobility support when both types of nodes are mobile is still an open challenge. It can be seen that there is still great deal of potentiality for addressing some other networking aspects. These include the support of effective intra-routing in a hierarchical multisink topology, which is the focus in this research work.

The evaluation process in some of the reviewed proposals carried out in different simulation environments. The Cooja simulator was a common choice among different ones [16-17, 23-24, 26-28] whereas the NS-2 was only used in [22]. More realistic and practical experimental evaluations were also conducted over real-life testbeds for the testing of the research works in [19-20,29]. The focus in most of these different works was on examining link reliability measurements such as PDR and end-to-end delay. Protocol performance measures such as control overhead and network hop count rate were also considered only in [26]. A different evaluation approach was taken by [25] in which the objective was to establish a baseline for RPL protocol timing settings.

## 4. MULTISINK RPL SOLUTION

Extending RPL with multisink support in a single DODAG requires inter-sink routing solution for effective forwarding of data traffic among the nodes joining different sinks. The approach proposed in this paper is to have prefix-based and hierarchical network topology establishment with aggregated inter-sink routing. The focus of this work is on the RPL storing mode which enables packet routing through any node in the DODAG in addition to the route.

The proposed solution requires each DODAG to have a main sink of ID 0 and a main prefix P0 advertised to its Network Segment (NS0). Each other sink attaches to NS0 as a leaf node. This could be via an additional physical interface or a logical one. It also must advertise a sub-prefix of P0 to its NS. A sink can attach to NS0 through S0 or any of its nodes. It is also possible for a node joining a NS of another sink to attach to NS0 through S0 or any of its nodes. Figure 2 presents all these supported cases in a simple multisink RPL setup of a single DODAG and 4 sink nodes. In this setup:

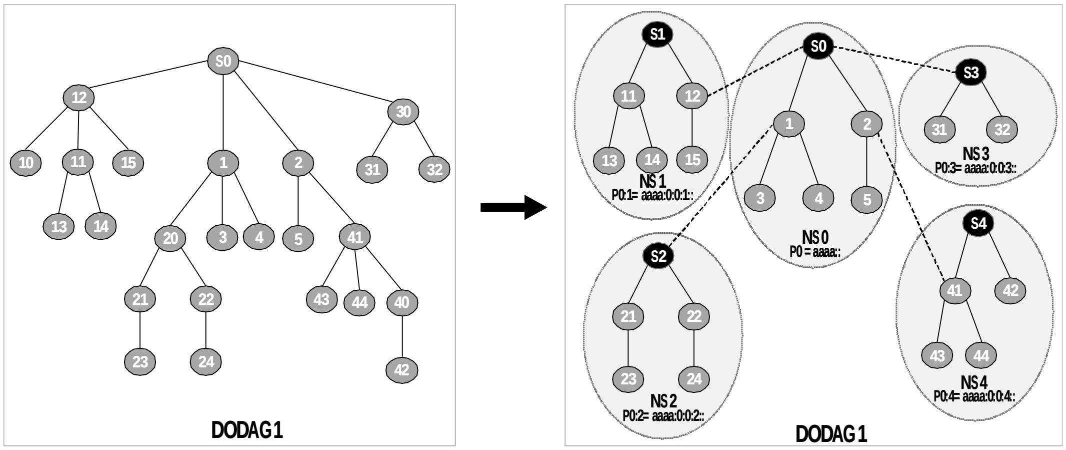

**Figure 2:** RPL Network Architecture

- S0 is the main sink of this DODAG and advertises the main prefix P0 to NS0
- The case of sink S3 represents the simple scenario as it attaches to the main NS (NS0) through the main sink (S0) and advertises the sub-prefix P0:3
- In the case of sink S2, it is unable to connect to S0 directly but attaches to NS0 through one of its RPL node (Node 1) and advertises the sub-prefix P0:2
- In another case, one of the nodes (Node 12) in NS1 of sink S1, which has a prefix of P0:1, attaches to NS0 through S0
- The last case illustrates that a node (Node 41) in NS4 of sink S4 advertising P0:4 can attach to NS0 through one of its RPL node (Node 2)

In the first case, as shown in Figure 3, sink S3 firstly receives a DIO message from S0. The message indicates the DODAG ID of 1 and contains the prefix aaaa:: in its Prefix Information Option (PIO). Upon that, sink S3 attaches to S0 and install a route for the aggregated prefix aaaa::/48 (via the main sink S0) in its routing table. A DAO message is then sent to S0 indicating the same DODAG ID and containing the prefix aaaa:0:0:3:: in a Target Option (TO). The first bit of the unused flag field of TO is set to indicate that this is prefix for another sink. Once received by S0, the DAO message is processed and route to the prefix aaaa:0:0:3::/64 (via S3) is installed in the routing table of S0.

At sink S2, a DIO message is received from Node 1 which is a node in NS0 and attached to S0. The message indicates the DODAG ID of 1 and contains the prefix aaaa:: in its Prefix Information Option (PIO). Sink S2 attaches to NS0 and installs a route for the aggregated prefix aaaa::/48 (via Node 1) in its routing table. S2 then sends a DAO message containing the prefix aaaa:0:0:2:: in a TO and the flag bit is set to 1. Upon receiving it at Node 1, a route to the prefix aaaa:0:0:2::/64 (via S2) is installed in its routing table. Node 1 then copies the received TO in a new DAO message to be sent to S0. Then, a route to the prefix aaaa:0:0:2::/64 (via Node 1) is installed in S0's routing table. Data traffic destined to any node in NS2 is then forwarded by Node 1 directly or through S0 (via Node 1) using the installed routes in their routing tables. An example in which a data packet is sent from Node 2 to Node 21 is shown in Figure 4. The packet would firstly be forwarded via Node 2's default route to S0. The packet is then forwarded by S0 to Node 1, as indicated by the already installed route in S0's routing table. Node 1 would then use the entry for the prefix aaaa:0:0:2::/64 in its routing table to forward the packet to S2 which then forwards it to the destination node (Node 21) using the normal RPL entries in its routing table.

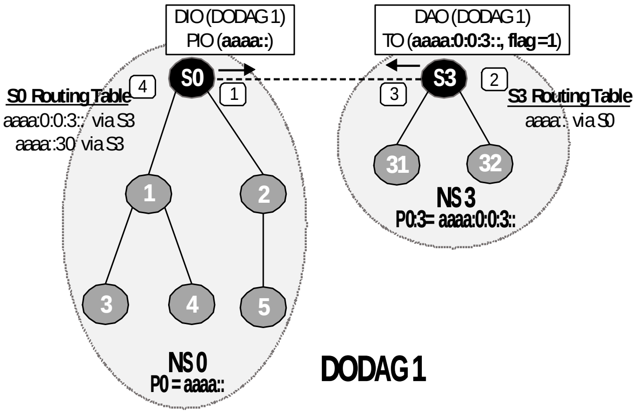

**Figure 3:** The Sink-to-Sink Intra-Routing Case

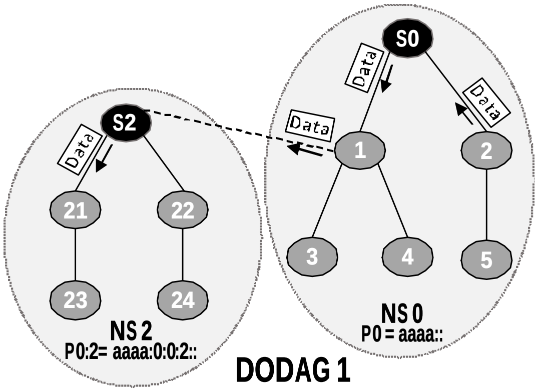

**Figure 4:** Data Traffic Routing Example

In the third case, Node 12 is a node in NS1 and associated with sink S1. After receiving a DIO message from S0, Node 12 obtains the advertised prefix in the message's PIO and install a route for the aggregated prefix aaaa::/48 (via S0) in its routing table. Node 12 then sends a DAO message to S1 after including a TO containing the prefix aaaa:: and a set flag bit. A route to the aggregated prefix aaaa::/48 (via Node 12) is then installed in the routing table of S1. Node 12 also sends a DAO message, indicating the same DODAG ID and containing the prefix aaaa:0:0:1:: in a TO with a set flag bit, to S0. Once received by S0, a route to the prefix aaaa:0:0:1::/64 (via Node 12) is installed in the S0's routing table. NS1 Data traffic destined to any node in NS0 would then be forwarded by Node 12 directly to S0 or through S1 via Node 12 using the installed routes in their routing tables.

In the last case, Node 41 which is a node in NS4 receives a DIO message from Node 2 that is a node in the NS0 as presented in Figure 5. Node 41 obtains the advertised prefix in the message's PIO and install a route for the aggregated prefix aaaa::/48 (via S0) in its routing table. Node 41 then sends a DAO message to S4 after containing the prefix aaaa:: into a TO with a set flag bit. A route to the aggregated prefix aaaa::/48 (via Node 41) is then installed in the routing table of S4. Node 41 also sends a DAO message, indicating the same DODAG ID and containing the prefix aaaa:0:0:4:: in a TO with a set flag bit, to Node 2. Once received by Node 2, a route to the prefix aaaa:0:0:4::/64 (via Node 41) is installed in Node 2's routing table. Node 2 then copies the received TO into a new DAO message to be sent to S0. NS4 Data traffic destined to any node in NS0 would then be forwarded by Node 41 directly or through S4 via Node 41 to Node 2, which then forward it over the normal RPL routes in NS0.

## 5. EXPERIENTIAL EVALUATION SETUP

Contiki OS is one of the common operating system for small-sized and limited-resources IoT devices. The network protocol stack of Contiki OS implements IP networking and supports IPv6 communications. It implements 6LowPAN and RPL for IP adaptation and network routing, respectively. The Contiki OS implementation [30] provides a simulation tool, namely the Cooja Simulator, which effectively facilitates the evaluation of wide range of IoT setups. It enables creating and running emulations of various IoT scenarios using different types of simulated IoT nodes that run the real Contiki OS implementation. In this work, Contiki version 3.0 and its Cooja simulator were used for the implementation and evaluation of the proposed solution.

First, The RPL code in the Contiki OS implementation was modified to implement the multisink RPL solution as described in the previous section. The current implementation supports the two default RPL Objective Functions. In this experiment, MRHOF was configured with its default routing metric, namely the Expected Transmission Count (ETX).

Then, an RPL instance with one DODAG was implemented using the Cooja Simulator to run different single and multi-sinks scenarios. As presented in Figure 6, the simulation setup consists of a total of 22 simulated nodes among of which are one or multiple sink nodes. In this setup, all the supported multisink cases are implemented similar to the setup shown in Figure 2. All the RPL nodes were emulated as Sky Mote devices. A UDP server was run by each of the sink nodes whereas the sensor nodes run different UDP client programs which periodically send IoT data every ±10 seconds. In a simulation area of about 200×200m, random

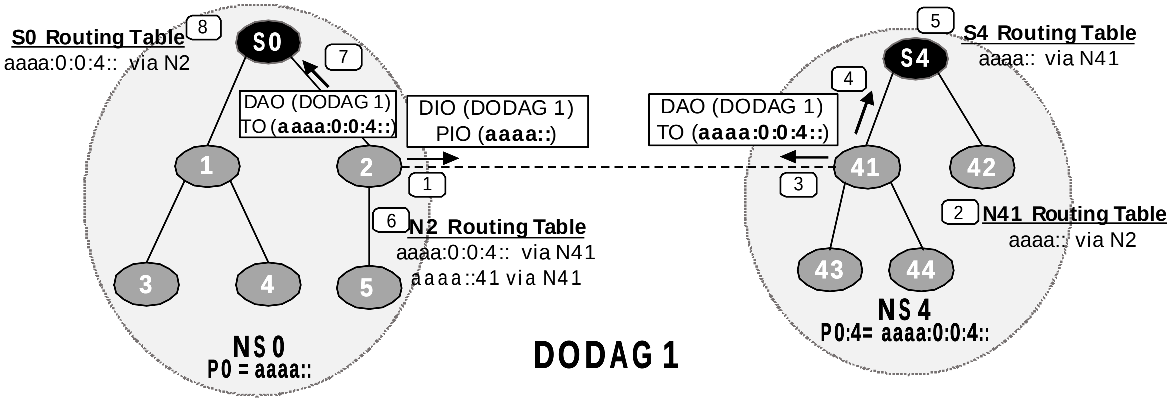

**Figure 5:** The Node-to-Node Intra-Routing Case

placement of the different nodes was adopted. Each node has a communication range of 25m and interference range of 50m. A multihop network topology was formed among all the simulated nodes.

The experimental evaluation started with a simulation scenario running the original RPL implementation with a single-sink setup. The resulted network performance in this original RPL setup was taken as a comparison baseline. Then, a number of simulation scenarios were carried out using the implemented multisink RPL solution. In each of these scenarios, the number of sink nodes was varied. Each scenario was run 10 times to take the average of the results afterwards. a simulation time of 15 minutes was set for each of these runs.

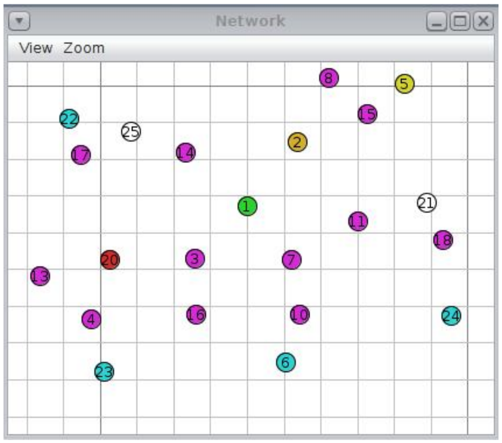

**Figure 6:** The Multisink Simulation Setup

The evaluation was based on comparing overall network performance in different multisink RPL setups to that in a single-sink RPL setup. In this work, the considered measures for performance evaluation were average end-to-end delay, Packet Delivery Ratio (PDR), and hop count. The calculation of the average end-to-end delay was based on the average time for all packets to reach the sink UDP server. The average PDR was taken as the ratio of total received packets at the UDP server to the total transmitted ones by the UPD senders.

## 6. RESULTS AND DISCUSSION

Figure 7 shows the hop count to the sink node for each of the considered scenarios. It can be seen that the topology was established with high number of hops among the nodes and the single sink node in the original RPL scenario. Increasing the number of sink nodes resulted in a noticeable decrease of the overall hop count. Using four sink nodes allowed for reducing the number of nodes that are 3 hops away by 58% and increasing those that are 1 hop away by 75%. It can also be seen that only one node stayed 3 hops away whereas the rest became at most 2 hops away when having 5 sink nodes.

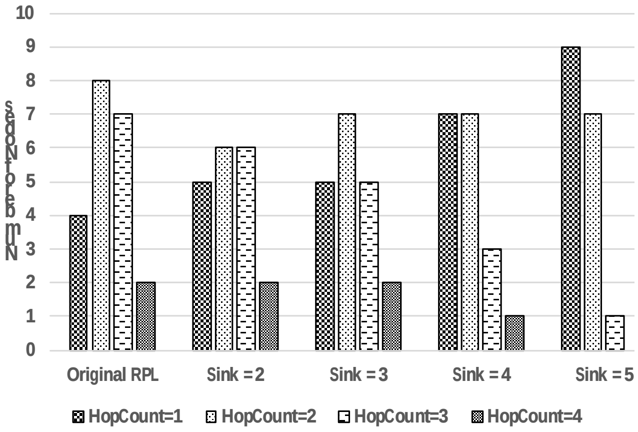

**Figure 7:** Hop Count results

The PDR results presented in Figure 8 indicate how increasing the number of sink nodes would result in better overall network performance. That is, increasing the sink nodes in the network alleviates the load on the bottleneck nodes at the higher level of the network topology. Adding only one additional sink node to the single-sink network setup resulted in an increase of about 5% in PDR. For the case when having 4 and 5 sink nodes, higher PDR was achieved with an increase of about 11 and 15%, respectively, compared to the original RPL scenario. Overall, the PDR improved by approximately 3-5% each time a sink node was added to the RPL network.

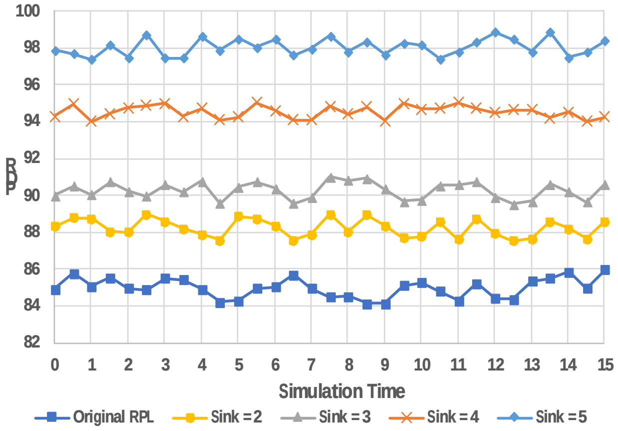

**Figure 8:** PDR results

Another important evaluation measure in our context is the delay experienced during local communications among the nodes associated to different sink nodes. This is important to understanding the effect of the proposed solution on the overall performance of the network. Table 2 presents the collected results of the overall average delay in milliseconds for selected nodes while considering the different cases. In the original RPL setup, there was no inter-sink communication, thus low overall delay was experienced by the different nodes. However, the multisink solution was able to maintain very low increase in the overall delay for the local communication when data was communicated over inter-sink network paths. At most, in the case of having 5 sink nodes, an increase of less than 2.5% was experienced by all the different local communication as shown in Table 2.

Table 2.The overall average delay considering the different cases

| Scenario | Node22-Node5 | Node24-Node6 | Node23-Node22 | Node5-Node23 |
|---|---|---|---|---|
| Original RPL | 112.14 | 110.93 | 113.07 | 113.10 |
| Sink = 2 | 112.76 | 110.98 | 113.09 | 113.96 |
| Sink = 3 | 114.01 | 111.03 | 114.71 | 114.04 |
| Sink = 4 | 114.46 | 112.27 | 114.92 | 115.16 |
| Sink = 5 | 114.88 | 112.87 | 115.28 | 115.57 |

## 7. CONCLUSION

The flexibility and versatility of the RPL routing model is evident. Advance networking supports such as multi-sink support can be easily incorporated into the basic RPL framework. The proposed multisink solution in this paper enables prefix-based and hierarchical network topology establishment with aggregated inter-sink routing. This resulted in a feasible and effective method to realize more routing optimization and improve network performance. As the experimentation results showed, RPL multi-sink support provides conspicuous improvement on network performance in terms of less hop count and better PDR, in addition to maintaining acceptable end-to-end delay. However, this solution would open the doors for addressing further RPL advancements in a future work. These include load balancing, congestion control, and QoS management. These would be feasible and viable considerations for reviving the capacity for more applicable and efficient IoT applications.

## REFERENCES

1. N. Kushalnagar, G. Montenegro, J. Hui, and D. Culler, “**Transmission of IPv6 Packets over IEEE 802.15.4 Networks**,” IETF RFC 4944, September 2007.
2. J. Hui and P. Thubert, “**Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks**,” IETF RFC 6282, September 2011.
3. T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik,J. P. Vasseurand, and R. Alexander, “**RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks**,” IETF RFC 6550, March 2012.
4. J. Vasseur, M. Kim, K. Pister, N. Dejean, and D. Barthel, “**Routing Metrics Used for Path Calculation in Low-Power and Lossy Networks**,” IETF RFC 6551, March 2012.
5. P. Thubert, “**Objective Function Zero for the Routing Protocol for Low-Power and Lossy Networks (RPL)**,” IETF RFC 6552, March 2012 https://doi.org/10.17487/rfc6552
6. O. Gnawali and P. Levis, “**The Minimum Rank with Hysteresis Objective Function**,” IETF RFC 6719, September 2012.
7. P. Levis, T. Clausen, J. Hui, O. Gnawali, and J. Ko, “**The Trickle Algorithm**,” IETF RFC 6206, March 2011. https://doi.org/10.17487/rfc6206
8. Masdari, M. and Naghiloo, F., 2017. **"Fuzzy Logic-Based Sink Selection and Load Balancing in Multi-Sink Wireless Sensor Network**s". Wireless Personal Communications, 97(2), pp.2713-2739.
9. S. Manishankar, P. R. Ranjitha and T. M. Kumar, "**Energy efficient data aggregation in sensor network using multiple sink data node**," 2017 International Conference on Communication and Signal Processing (ICCSP), Chennai, 2017, pp. 0448-0452. https://doi.org/10.1109/ICCSP.2017.8286397
10. S. Anandalatchoumy and G. Sivaradje, "**Multiple Mobile Sink Architecture Based Energy Efficient Routing Protocol for Underwater Wireless Sensor Networks**," 2018 IEEE International Conference on System, Computation, Automation and Networking (ICSCA), Pondicherry, 2018, pp. 1-6.
11. Amer O. Abu Salem, 2019. "**A Novel Leach Routing Protocol using Genetic Algorithm for Wireless Sensor Networks**", International Journal of Advanced Trends in Computer Science and Engineering (IJATCSE), Volume 8, No.5, September - October 2019. https://doi.org/10.30534/ijatcse/2019/37852019
12. B.V.Shruti1, M.N.Thippeswamy, Venkatesh K, 2019, "**Energy Efficient Medium Access Control Protocols for Wireless Sensor Networks – A Survey**", International Journal of Advanced Trends in Computer Science and Engineering (IJATCSE), Volume 8, No.1.5, 2019. https://doi.org/10.30534/ijatcse/2019/6381.52019
13. Sha, C., Qiu, J.M., Li, S.Y., Qiang, M.Y. and Wang, R.C., 2016. "**A type of low-latency data gathering method with multi-sink for sensor networks**". Sensors, 16(6), p.923.
14. J. P. J. Peixoto and D. G. Costa, "**QoE-aware multiple sinks mobility in wireless sensor networks**," 2015 7th International Conference on New Technologies, Mobility and Security (NTMS), Paris, 2015, pp. 1-4.
15. G. Yestemirova and S. Saginbekov, "**Efficient Data Aggregation in Wireless Sensor Networks with Multiple Sinks**," 2018 IEEE 32nd International Conference on Advanced Information Networking and Applications (AINA), Krakow, 2018, pp. 115-119.
16. M. O. Farooq, C. J. Sreenan, K. N. Brown, and T. Kunz, “**RPL-based Routing Protocols for Multi-sink Wireless Sensor Networks**,” in Proc. 11th IEEE International Conference on Wireless and Mobile

Computing, Networking, and Communications (WiMob), Abu Dhabi, Oct 2015, pp. 452-457. https://doi.org/10.1109/WiMOB.2015.7347997

17. M.O. Farooq, C. J. Sreenan, K. N. Brown, and T. Kunz, "**Design and Analysis of RPL Objective Functions for Multi-Gateway Ad-Hoc Low-power and Lossy Networks**," Ad Hoc Networks, vol. 65, no. 3, pp. 78-90, Oct 2017.
18. D. Carels, N. Derdaele, E. De Poorter, W. Vandenberghe, I. Moerman, and P. Demeester, "**Support of Multiple Sinks via a Virtual Root for the RPL Routing Protocol**," EURASIP Journal on Wireless Communications and Networking, Vol. 2014, no. 1, pp. 91-113, June 2014.
19. QD. Nguyen, J. Montavont, N. Montavont, and T. Noël, "**RPL Border Router Redundancy in the Internet of Things**," in Proc. International Conference on Ad-Hoc Networks and Wireless, Springer, Cham, July 2016, pp. 202-214.
20. L. Deru, S. Dawans, M. Ocaña, B. Quoitin, and O. Bonaventure, "**Redundant Border Routers for Mission-Critical 6LoWPAN Networks**," in Proc. 5th Workshop on Real-World Wireless Sensor Networks (REALWSN), Springer, Como, Sep 2013, pp. 195-203. https://doi.org/10.1007/978-3-319-03071-5_20
21. Wang, J. and Chalhoub, G., 2019. "**Mobility support enhancement for RPL with multiple sinks**". Annals of Telecommunications, pp.1-14.
22. P. Thulasiraman, "**RPL Routing for Multigateway AMI Networks Under Interference Constraints**," in Proc. IEEE International Conference on Communications (ICC), Budapest, June 2013, pp. 4477-4428.
23. M. M. Khan, M. A. Lodhi, A. Rehman, and F. B. Hussain, "**A Multi-Sink Coordination Framework for Low Power and Lossy Networks**," in Proc. International Conference on Industrial Informatics and Computer Systems (CIICS '16), Sharjah, March 2016, pp. 1-5.
24. M. M. Khan, M. A. Lodhi, A. Rehman, A. Khan, and F. B. Hussain, "**Sink-to-Sink Coordination Framework Using RPL: Routing Protocol for Low Power and Lossy Networks**," Journal of Sensors, Vol. 2016, pp. 1-11, 2016.
25. K. Andrea and R. Simon, "**Design and Evaluation of an RPL-based Multi-Sink Routing Protocol for Low-Power and Lossy Networks**," in Proc. 18th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM), Cancun, Nov 2015, pp. 141-150. https://doi.org/10.1145/2811587.2811614
26. H. Lamaazi, N. Benamar, and A. J. Jara, "**RPL-based Networks in Static and Mobile Environment: A Performance Assessment Analysis**," Journal of King Saud University - Computer and Information Sciences, vol. 30, no. 3, pp. 320-333, July 2018.
27. Shabbir, G., Akram, A., Iqbal, M.M., Jabbar, S., Alfawair, M. and Chaudhry, J., 2018. "**Network Performance Enhancement of Multi-sink Enabled Low Power Lossy Networks in SDN Based Internet of Things**". International Journal of Parallel Programming, pp.1-32. https://doi.org/10.1007/s10766-018-0620-8
28. Sebastian, A. and Sivagurunathan, S., 2018. "**Multi Sink RPL based Internet of Things for Emergency Response in Smart Cities**." International Journal of Pure and Applied Mathematics, 118(18), pp.2875-2881.
29. Ara Miran1, Govand Kadir2, 2019, " **Enhancing AODV routing protocol to support QoS**", International Journal of Advanced Trends in Computer Science and Engineering, Volume 8, No.5, September - October 2019.
30. A. Dunkels, B. Gronvall, and T. Voigt, "**Contiki- a lightweight and flexible operating system for tiny networked sensors**," In proc. The 29th Annual IEEE International Conference on Local Computer Networks, Tampa, FL, USA, Nov 2004, pp. 455-462.