

RESEARCH

Open Access

# Routing protocol for Low-Power and Lossy Networks for heterogeneous traffic network



Arslan Musaddiq<sup>†</sup>, Yousaf Bin Zikria<sup>†</sup>, Zulqarnain and Sung Won Kim<sup>\* ID</sup>

## Abstract

In a real network deployment, the diverse sensor applications generate a heterogeneous traffic pattern which may include basic sensing measurements such as temperature readings or high-volume multimedia traffic. In a heterogeneous traffic network, the two standardized objective functions (OFs), i.e., objective function zero (OF0) and the Minimum Rank with Hysteresis Objective Function (MRHOF) for routing protocol for Low-Power and Lossy Networks (RPL) perform poor routing decisions by selecting an already congested parent node and cause more re-transmissions across the network. Therefore, careful consideration is required in designing a new OF for heterogeneous traffic scenarios. In this study, we examine the RPL protocol under a heterogeneous traffic pattern and proposed a new protocol based on queue and workload-based condition (QWL-RPL). The aim of the proposed protocol is to achieve a reliable path with better overall performance. The proposed OF model considers the link workload in addition to mapping the congestion status of the node using the packet queue. We implement the proposed routing model in the Contiki operating system (OS) Cooja environment to compare with the existing technique. The simulation results show that QWL-RPL can improve the performance of a heterogeneous traffic network as compared with both OF0 and MRHOF, specifically in terms of the amount of overhead, packets reception ratio (PRR), average delay, and jitter. Final results indicate that on average, there is a 5%–30% improvement in PRR, 25%–45% reduction in overheads, 12%–30% reduction in average delay, and 20%–40% reduction in jitter.

**Keywords:** Internet of Things, Routing, Objective function, Low-Power and Lossy Network, RPL

## 1 Introduction

The Internet of Things (IoT) is a large system of interconnected heterogeneous devices to sense our physical environment and communicate the sensed information. Each sensor detects its vicinity and transmits the sensed information to the sink node via its neighboring nodes using multi-hop communication [1]. The practical applications of such a system include smart cities [2] [3], industrial and building automation, disaster management [4], smart grid, and smart healthcare [5]. A sensor discovers its neighbors, builds the topology, and routes the sensed data. An IoT system requires an efficient networking mechanism to facilitate such an interconnected heterogeneous flow of traffic. An efficient routing mechanism must consider these device characteristics and constrained resources. Furthermore, the

IoT sensors usually face unfavorable environmental factors. Thus, designing of such routing protocols is a complicated task due to the resource limitations of these devices, such as limited energy, limited memory, and limited processing power [6]. For any pair of communicating nodes, a reasonable goal for such a protocol is to improve communication performance while maintaining energy consumption at a minimum level. However, achieving this goal requires careful consideration of the lossy nature of the network, fluctuating traffic patterns, varying link qualities, routing loops, and convergence time [7].

Several routing protocols have been proposed for these Low-Power and Lossy Network (LLN) devices; these include protocols such as Tiny ad hoc on-demand distance vector (TinyAODV) [8], hierarchical routing over 6LoWPAN (hilow) [9], 6LoWPAN ad hoc on-demand distance vector routing (LOAD) [10], dynamic MANET on-demand for 6LoWPAN (DYMO-low) [11], collection tree protocol (CTP) [12], and hybrid routing protocol for lossy

\* Correspondence: [swon@yu.ac.kr](mailto:swon@yu.ac.kr)

<sup>†</sup>Arslan Musaddiq and Yousaf Bin Zikria contributed equally to this work. Department of Information and Communication Engineering, Yeungnam University, 280, Daehak-Ro, Gyeongsan-si, Gyeongsangbuk-do 38541, South Korea

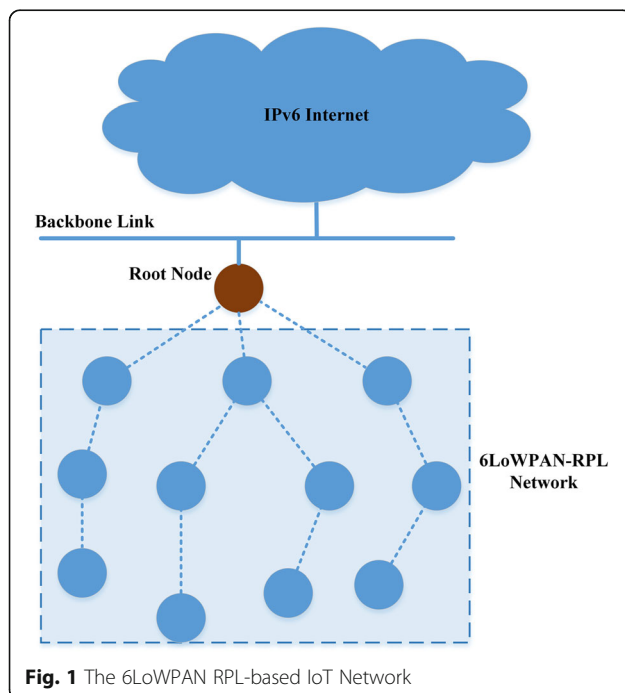
and low power networks (Hydro) [13]. One of the promising routing protocols that provides IPv6 connectivity to the LLN devices is called the routing protocol for Low-Power and Lossy Networks (RPL), as proposed by the International Engineering Task Force (IETF), ROLL (Routing Over Low-Power and Lossy Networks) working group [14]. The RPL protocol is based on the IPv6 low-power wireless personal area network (6LoWPAN) which is connected to the IP network through the sink node (Fig. 1) [15]. The RPL protocol has been proposed for a wide range of networking environments such as home automation, building automation, industrial automation, urban routing, and the smart grid [16] [17]. The RPL is being utilized commercially by Cisco for the field area network (FAN) and smart grids (CG-Mesh) to enable the advanced meter infrastructure (AMI) [18].

The RPL protocol is a distance vector proactive routing protocol that creates a tree-like routing topology called the destination-oriented directed acyclic graph (DODAG), rooted towards one or more nodes called the root node or sink node. The directed acyclic graphs (DAGs) are created based on a user-specified specific objective function (OF). The OF defines the method to find the best-optimized route among the number of sensor devices [14]. The IETF ROLL working group standardized the objective function zero (OF0) [19] and the minimum rank with hysteresis objective function (MRHOF) as default routing metrics defined in RFC 6719 [20]. The OF0 finds the shortest path to the sink node by selecting the candidate parent node with

minimum rank in terms of the distance from the sink (i.e., its position in the routing tree). The MRHOF finds the routes through the sensor nodes that minimize the link cost associated with the routes. It selects the new routing path if the cost associated with it is less than the current path cost by a given threshold value. This is known as ‘hysteresis.’ As prescribed in the standard, MRHOF utilizes the expected transmission count (ETX) metric which calculates the link quality. Furthermore, ETX considers link-layer congestion but does not reflect node level congestion. Therefore, selecting a routing path based on the smallest hop count and link quality in a heterogeneous traffic environment does not lead to an efficient load-balancing solution. The nodes closer to the sink node suffer from packet loss due to a high relay burden in a dense networking environment. The chosen parent node in RPL can have multiple child nodes; consequently, the overloaded preferred parent becomes prone to failure because its energy drains much faster than other nodes. The inefficient OFs lead to building a routing topology that experiences an excessively unbalanced load and energy distribution, particularly for those nodes that are closest to the sink node.

The diverse applications of LLNs include scenarios ranging from basic temperature measurements to high-volume multimedia services that require efficient communication support. The LLN heterogeneous traffic environment suffers from severe congestion and packet loss by not utilizing the full network capacity. A high relay burden, unbalanced load, and limited resources eventually lead to node failure. The ETX-based node broadcasts probe packets at time intervals to assess the link quality. The receiving node re-broadcasts the probe packet which further increases the network congestion. If there is a node failure, RPL initiates two types of repairs: local repair and global repair. In the local repair scenario, a child node routes the packets through its sibling node or the child node switches to the parent node. The global repair is initiated by a gateway or sink node. In both cases, the network incurs overall delay and additional control overhead, which, in turn, becomes a detriment to the overall network performance. When the number of failed nodes increases significantly, the network is split in such a way that communicating with the sink node is not possible through any path, resulting in a non-operational network [14, 20].

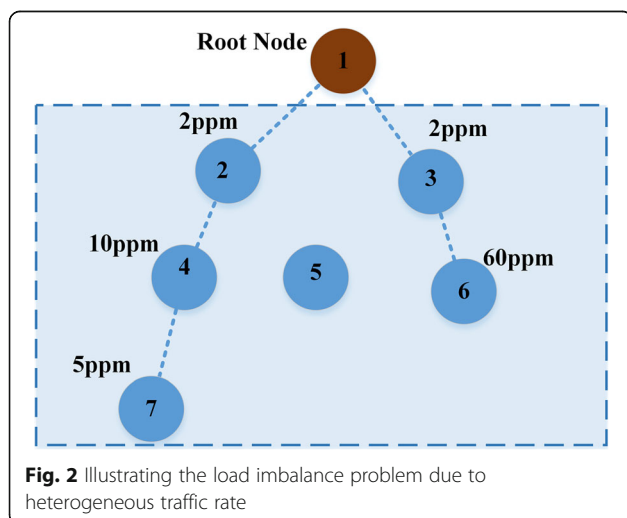
The generic definition of heterogeneity is a lack of uniformity. In terms of heterogeneous traffic load, some of the nodes are traffic-intensive while others generate traffic with a low traffic rate. In a normal scenario, the sensor nodes form a tree topology to transfer data towards the root node. The traffic flow is homogeneous with a constant period. In this case, the resulting traffic pattern is predictable and load imbalance is unlikely. In the heterogeneous case, the transmission interval, as well as



**Fig. 1** The 6LoWPAN RPL-based IoT Network

transmission load, differ in each node; thus, the load fluctuates unpredictably causing a significant load imbalance problem.

In a heterogeneous traffic load scenario, the load imbalance may not be due to the subtree size, as illustrated in Fig. 2, where node 5 has a choice to select the node 2 or node 3 as its parent node. As node 3 contain only one child node, node 5 is likely to select node 3 as its parent node. This scenario is likely to happen in the same traffic load case. However, in this network, the accumulated workload at node 3 is greater than that of node 2 due to the high packet generation rate of node 6, i.e., 60 ppm (packets per minute). Therefore, having more nodes in the routing subtree does not always mean that more traffic flows through that node in a heterogeneous traffic network. In [20], it is observed that parent nodes with large subtree sizes have large queue losses. However, in a heterogeneous traffic pattern, it may not always be true that a parent node with large subtree will have large queue losses. In heterogeneous traffic, the imbalanced loads and queue losses are primarily due to the unbalanced traffic generation as compared with an unbalanced subtree size. The ETX primarily reflects the link-level losses rather than node-level packet losses. On the contrary, the hop count only reflects the number of hops irrespective of traffic load and queue losses. In the same way, utilizing only the queue size as a decision metric does not truly reflect the load condition. The workload of each node is an important parameter in an uneven traffic load pattern. Similarly, the protocol design for such cases must consider latency or average end-to-end (E2E) delay and the amount of overhead as an important performance parameter. Therefore, it is necessary to consider a new routing metric that solves the load balancing problem in a heterogeneous traffic load scenario.



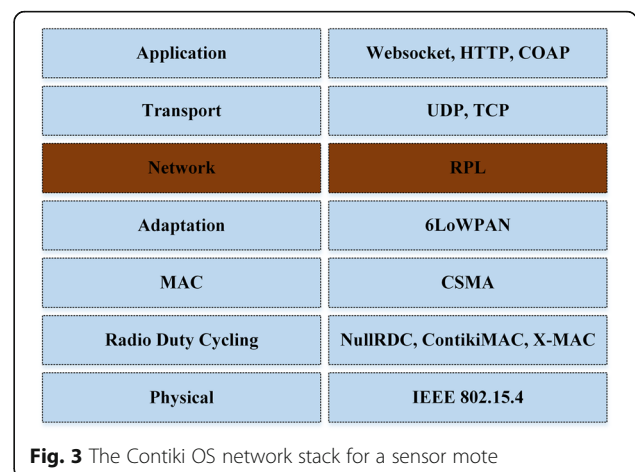
**Fig. 2** Illustrating the load imbalance problem due to heterogeneous traffic rate

Various routing metrics have been proposed for homogeneous traffic loads; these metrics are based on node and link-level attributes such as node residual energy [21], queue utilization [22], throughput, and latency for parent selection in a heterogeneous network. These routing metrics are explained in Section 2. The main contributions of our work are to fill the gap in the proposed OFs and provide a mechanism that achieves an improved packet reception ratio (PRR), average E2E packet delay, jitter, overall load balancing, and enhanced network lifetime while conforming to the RPL standard. To handle this issue, we propose an OF for a heterogeneous traffic load based on queue and workload (QWL-RPL) that provides an extension to the RPL. The proposed metric considers the load and congestion during uneven traffic distribution to make the best decision for route selection.

The remainder of the paper is organized as follows: Section 2 presents the related work and Section 3 discusses the QWL-RPL implementation and analysis. Section 4 describes the simulation method and experiment. The experimental results and analysis are presented in Section 5. Finally, the conclusion and future work are discussed in Section 6.

## 2 Related work

The RPL is standardized for LLNs as a proactive routing protocol that operates on top of the 802.15.4 physical (PHY) and media access control (MAC) layers (Fig. 3) [23–25]. As the sensor nodes are small and usually contain a limited battery as an energy source, the network protocol stack of the sensor nodes often includes a radio duty cycling (RDC) protocol such as NullRDC, ContikiMAC [26], and X-MAC [27]. NullRDC does not use framer functions and transmit/receive the data directly to the physical layer. ContikiMAC uses periodical wake-ups to listen for packet transmissions from neighbors whereas X-MAC protocol does not switch off its radio after packet transmission. These RDC protocols are all



**Fig. 3** The Contiki OS network stack for a sensor mote

implemented in the Contiki OS Cooja simulator [28]. The RDC protocol can increase energy efficiency by silencing the radio when there is no sensing activity. RPL uses a specific OF that defines the use of specific metrics for rank calculation to make an intelligent routing decision. It constructs a tree-like routing topology called DODAG. The primary objectives of the RPL protocol are to create an optimal DODAG and to adapt the topology with respect to various network situations. To manage various routing challenges, the routing metrics must consider different constraints and requirements of the LLN network. The RPL protocol does not specify any particular OF or routing metric to be used. Rather, it provides a great degree of flexibility for defining any OF based on network applications and design. In recent years, RPL has been studied in different contexts and for a variety of platforms; several protocols are presented to study the variations of RPL protocol. Various analyses are performed to critically analyze the reliability and performance of RPL variations. Traditionally, ETX in MRHOF and hop count in OF0 are provided by the RPL standard implementation. The OF0 implementation does not use any specific routing metric for rank calculation other than hop count. MRHOF accesses the link quality between the nodes to select the preferred parent node. This approach is more efficient and is favored in RPL models [19, 20].

The routing protocols discussed in this literature are primarily focused on increasing the network performance in terms of load balancing, congestion, and energy consumption. These protocols employ various routing metrics and scenarios to improve the network performance through the routing protocol. The authors in [22] present a queue-utilization-based RPL (QU-RPL) that evaluates the end-to-end delivery performance. This method balances the routing tree under heavy traffic scenarios. The proposed mechanism discusses the congestion and load balancing issue by considering the queue utilization factor in node selection. The queue utilization factor reflects the congestion state of the node during parent selection. The node chooses the parent that has less buffer occupancy and lower hop counts from the sink node. Bhandari et al. propose a congestion-aware routing protocol (CoAR) [29] that uses a multi-criteria decision-making approach.

CoAR utilizes the order preference by similarity to ideal solution (TOPSIS) [30] and combines multiple routing metrics for parent selection. The proposed OF is based on the queue utilization, ETX, remaining energy, and neighborhood index. The neighborhood index defines the ratio of the number of downstream neighbors to the total number of neighboring nodes.

Tang et al. also discuss the congestion avoidance mechanism and proposed a composite metric called CA-RPL [31]. The CA-RPL determines the average delay towards the sink node and computes the weight of each path. It also includes the rank of the node, ETX value, and the number of received packets. Taghizadeh et al. [32] propose a metric for the energy and packet loss problem under heavy traffic load scenarios. The rank of the node is computed based on the context of the node. The authors in [33] propose two MAC-aware routing metrics called R-metric and Q-metric. The R-metric enhances the ETX protocol by considering the packet drops due to MAC contention, and the Q-metric solves the load balancing problem by considering the power consumption during transmission and reception of the packets. Sanmartin et al. propose SIGMA-ETX [34] which considers the standard deviation of ETX values between each node. This protocol provides efficient performance in a dense network.

Another metric to achieve load balancing is named ETT-LB [35] which considers the transmission rate and size of the packet in addition to the expected delay time. Similarly, the authors in [36] proposed another load balancing technique called ALABAMO, which considers the node selection using both the ETX and the traffic load profiles of the nodes. Liu et al. also proposed an interesting load balancing mechanism based on the RPL protocol called LB-RPL [37]. It considers the workload differences and spreads the data traffic among multiple parent nodes. It leverages the concept of multiparent routing and utilizes all the potential parent nodes for next hop data forwarding. The data is distributed proportionally according to the measured link quality status.

The multipath routing approach is utilized to provide load balancing [38], reliability [39], and fault-tolerance [40]. Iova et al. propose a multi-parent protocol and presents the expected lifetime (ELT) [41] metric to estimate

**Table 1** ContikiRPL objective function parameters and their associated files

Contiki OS file	OF0 Parameters	MRHOF Parameters
rpl-conf.h	RPL_OF_OCP RPL_OCP_OF0 RPL_SUPPORTED_OF0 {&rpl_of0}	RPL_OF_OCP RPL_OCP_MRHOF RPL_SUPPORTED_OF0 {&rpl_mrhop} RPL_WITH_MC 1 RPL_DAG_MC_ETX (for ETX metric) RPL_DAG_MC_ENERGY (stub energy container)
rpl.h	RPL_OCP_OF0 1	RPL_OCP_MRHOF 1

**Table 2** ContikiRPL analysis of heterogeneous traffic load scenarios with OF0 and MRHOF

Traffic load (ppm)	No. of nodes	OF	Total pkts sent	Total pkts rcvd	Total packets lost	PRR %	Packets lost %	Avg. delay (ms)	Jitter (ms)	Total overheads	Network energy consumption (mJ)
1, 10, 30, 60	20 nodes	OF0	30,241	23,843	6398	78.84	21.16	33.76	14.14	60,472	3,745,690.47
1, 10, 30, 60	20 nodes	MRHOF	30,202	27,733	2469	91.83	8.17	27.53	8.5	17,007	3,744,494.26

the lifetime of the bottlenecked nodes. The proposed ELT metric considers the amount of traffic and link reliability to estimate the energy hotspots. Another energy-based routing metric for RPL is defined in [42] that uses the remaining energy as a main decision parameter for the next hop selection.

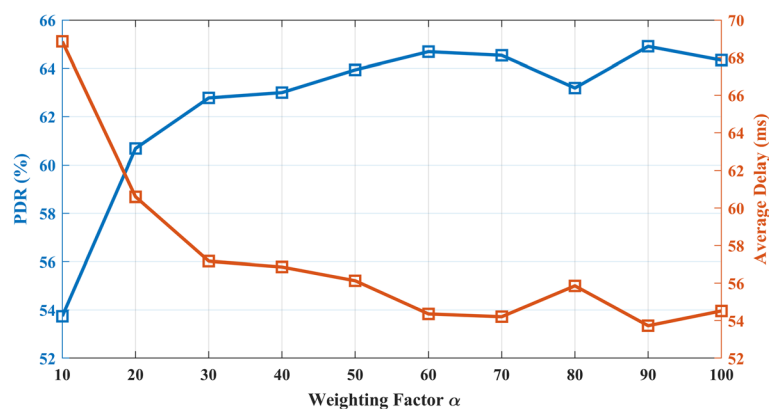
Zhao et al. define a region-based energy-efficient routing protocol called ER-RPL [43]. In ER-RPL, only a subset of the nodes participates in the route discovery which results in the reduction of the control overheads. Natanael et al. [44] also propose an energy-efficient and path-reliability-aware objective function (ERAOF) that also includes energy consumption as a routing metric to avoid the low-energy paths. Barbato et al. propose the resource-oriented and energy-efficient routing (ROEE) RPL protocol [45]. The ROEE RPL protocol retrieves resource information requested by the application layer and defines the routing path based on the resource requirements of the network. The proposed metric uses the energy consumption and battery index of each node.

In a similar way, the scalable context-aware objective function (SCAOF) [46] has been proposed which utilizes the weighted sum of three metrics, i.e., degree of node connectivity, node energy, and node location in the DODAG relative to the parent node. The diverse applications of IoT networks require numerous performance parameters. Thus, the authors in [47] combine multiple routing metrics which are the point-to-point delay, ETX, hop count, and battery energy level. Using these metrics, the proposed objective function is based on fuzzy logic.

Patrick et al. [48] also utilize the concept of fuzzy logic to combine several routing metrics that include ETX, delay, and energy cost of the path. Regarding multiple metrics, Walid et al. propose a new OF which is based on the non-linear length (NL-OF) [49]. The NL-OF is a greedy approach that considers any number of metrics and all input constraints for quality of service (QoS).

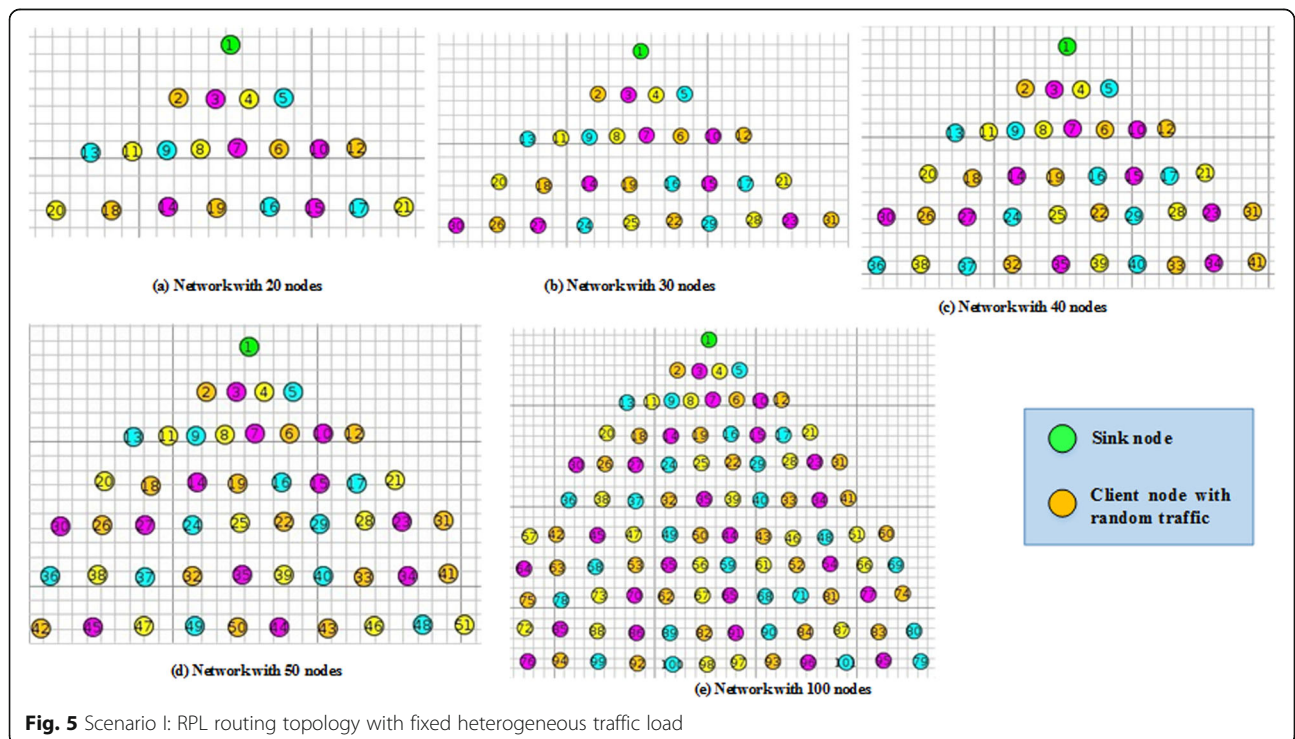
The authors in [50] use the residual energy depletion rate for the routing decision. Elnaz et al. [51] formulated a transmission power control technique to reduce power consumption and Barbato et al. [52] exploit the node energy status and the node's position in the network. Brachman et al. [53] provide the analysis of the OF0 and link quality OF (LLQ OF). The OF0 minimizes the hop count while LLQ OF uses link quality metric derived from the received signal strength indicator (RSSI) value to maintain the network stability. The bio-inspired particle multi-swarm optimization (PMSO) routing algorithm for multipath selection is proposed in [54]. PMSO achieves fast recovery from path failure by utilizing average energy consumption and average in-network delay as a routing objective function.

The optimization model proposed in [55] aims to reduce the expected end-to-end transmission time. The proposed method determines the weighted cumulative expected transmission time (WCETT) which depends on the number of links in the network. It then utilizes the optimization model to reduce the sum of all WCETT's while considering the bandwidth, flow control, and power control constraints.

**Fig. 4** PRR (percent) and average delay (ms) with varying weighting factor

**Table 3** Simulation model parameters and their values

Parameters	Value
Contiki OS version	Contiki 3.0
Mote device model	Z1 Zolertia
Z1 mote voltage	3 V
Z1 mote TX current	17.4 mA
Z1 mote RX current	18.8 mA
Z1 mote CPU idle current	0.426 mA
Z1 mote CPU power down current	0.020 mA
RTIMER_SECOND	32768 ticks per second
CC2420 radio CCA threshold	-45 dbm
Z1 RAM	8 KB
Z1 flash	92 KB
Packet size	127 bytes
Buffer occupancy	4 packets
uIP payload buffer size	140 bytes
Simulation time	3600 s
PHY & MAC protocol	802.15.4 with CSMA
Radio	CC2420
Number of nodes	20, 30, 40, 50, 100
Send interval	Scenario 1: 1 * clock second, 2 * clock second, 6 * clock second, 60 * clock second Scenario 2: random number between 1 and 15 * clock second/random 1-10 Random number seeds: new seed with respect to clock time each iteration
Workload measurement interval	Every 10 s
Script text analysis	Python 3.7



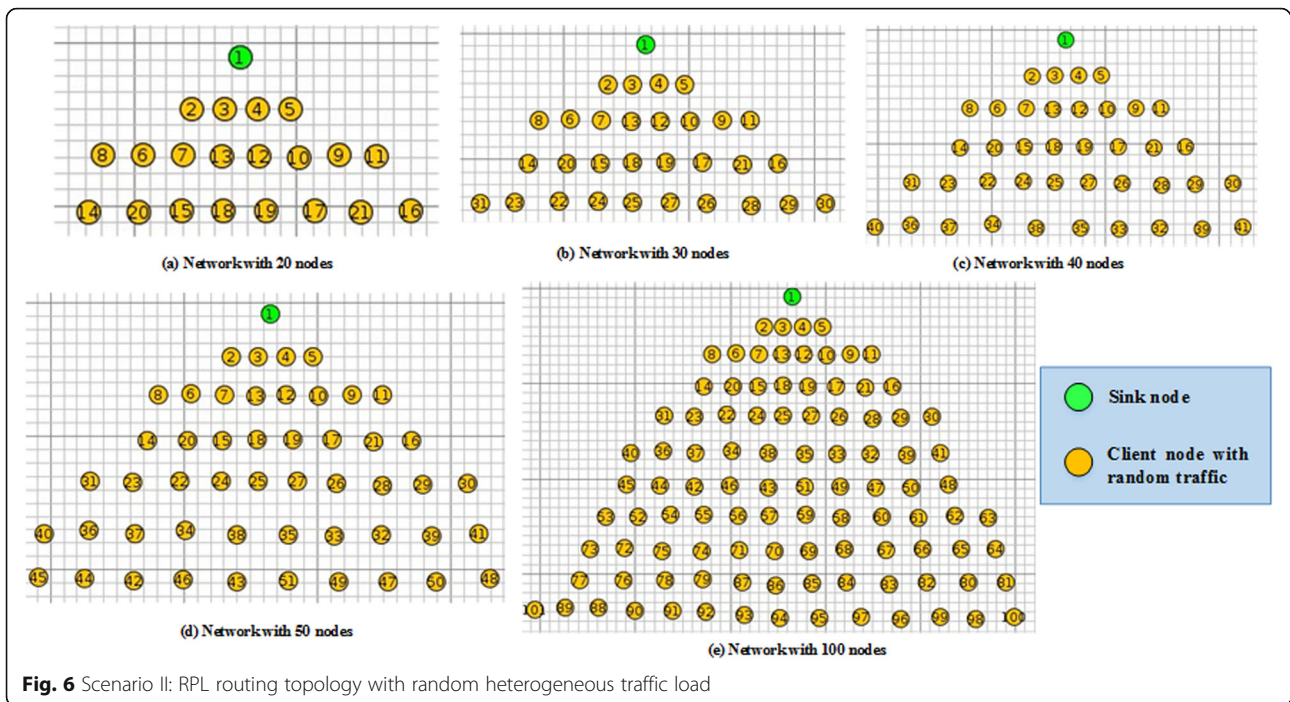


Fig. 6 Scenario II: RPL routing topology with random heterogeneous traffic load

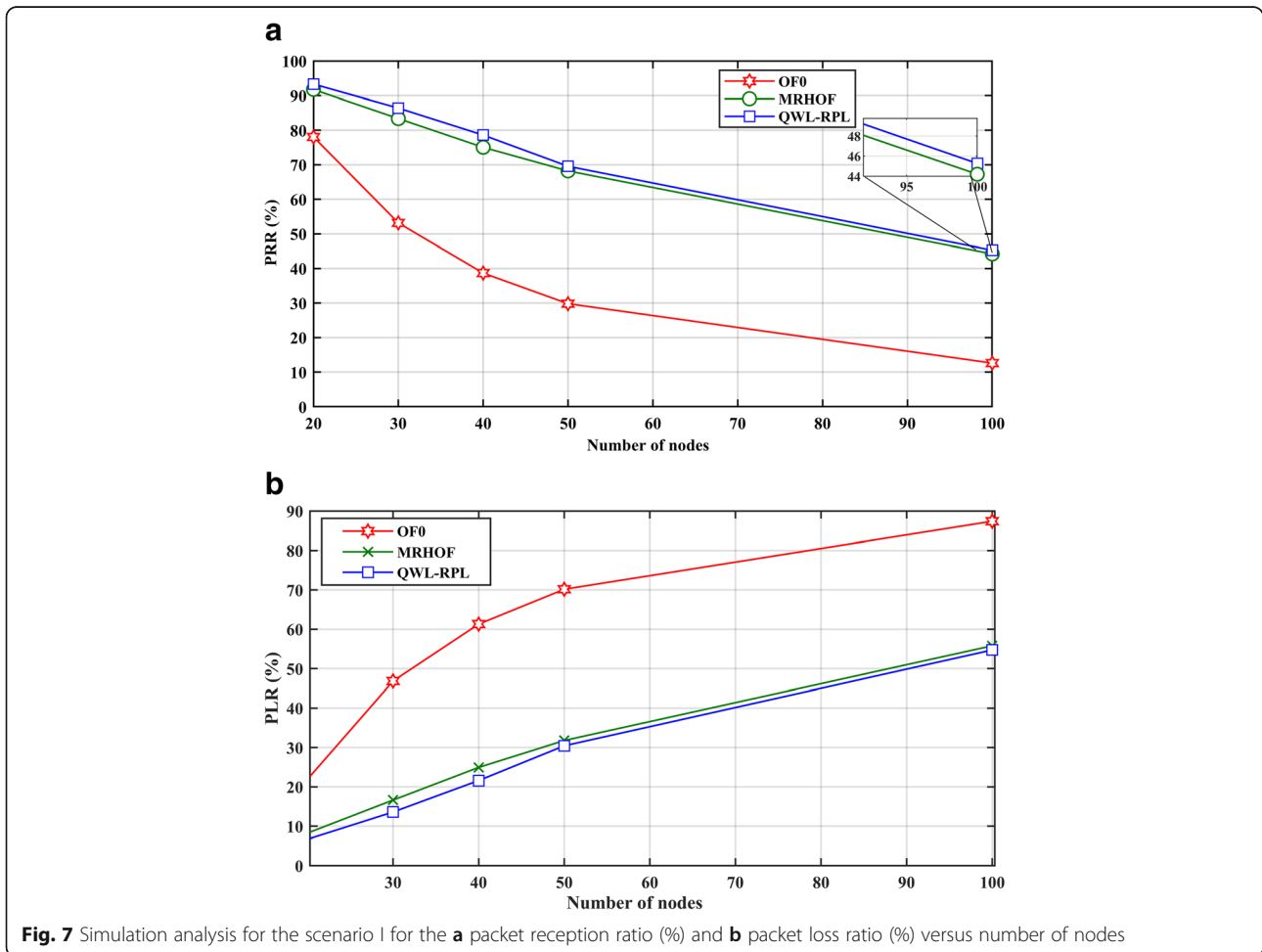


Fig. 7 Simulation analysis for the scenario I for the a packet reception ratio (%) and b packet loss ratio (%) versus number of nodes

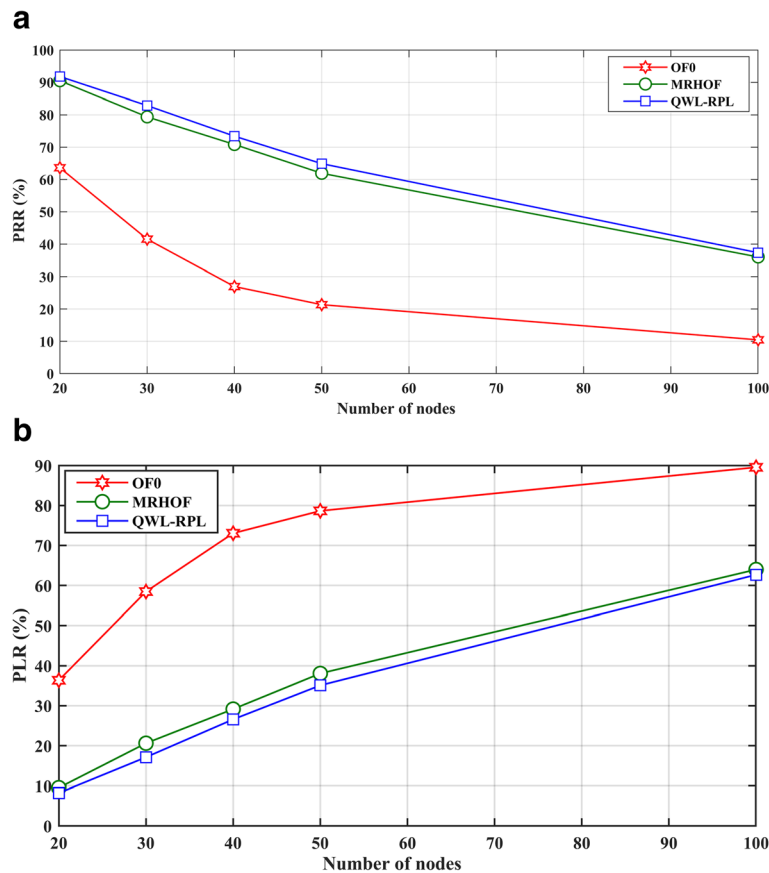


Fig. 8 Simulation analysis for the scenario II for the a packet reception ratio (%) and b packet loss ratio (%) versus number of nodes

### 3 QWL-RPL implementation and analysis

This section describes the QWL-RPL protocol which considers the load imbalance problem during a heterogeneous traffic scenario. We consider a network with two different types of traffic heterogeneities. The first scenario contains a network where nodes transmit with a fixed heterogeneous traffic pattern; the second scenario involves traffic generation randomness that varies with time.

#### 3.1 System model

Consider an LLN IoT scenario forming an RPL routing topology. The network comprises a multihop network

for data transmission from a child node to the root node, forming a DODAG routing graph. The network consists of a number of nodes  $N$  which are ranked in increasing order from the root node. The node model includes the Texas Instruments CC2420 radio transceiver. Based on their rank, these nodes are divided into sets of parent and child nodes  $N = P \cup C$ , where  $P$  represents the set of parent nodes ( $p_1, p_2, p_3, \dots, p_n$ ) and  $C$  represents the set of child nodes ( $c_1, c_2, c_3, \dots, c_n$ ). The candidate parent node is given a rank one if it is a root node. The sink node connects the LLN nodes to the IPv6 network forming a wide area network (WAN). Each node

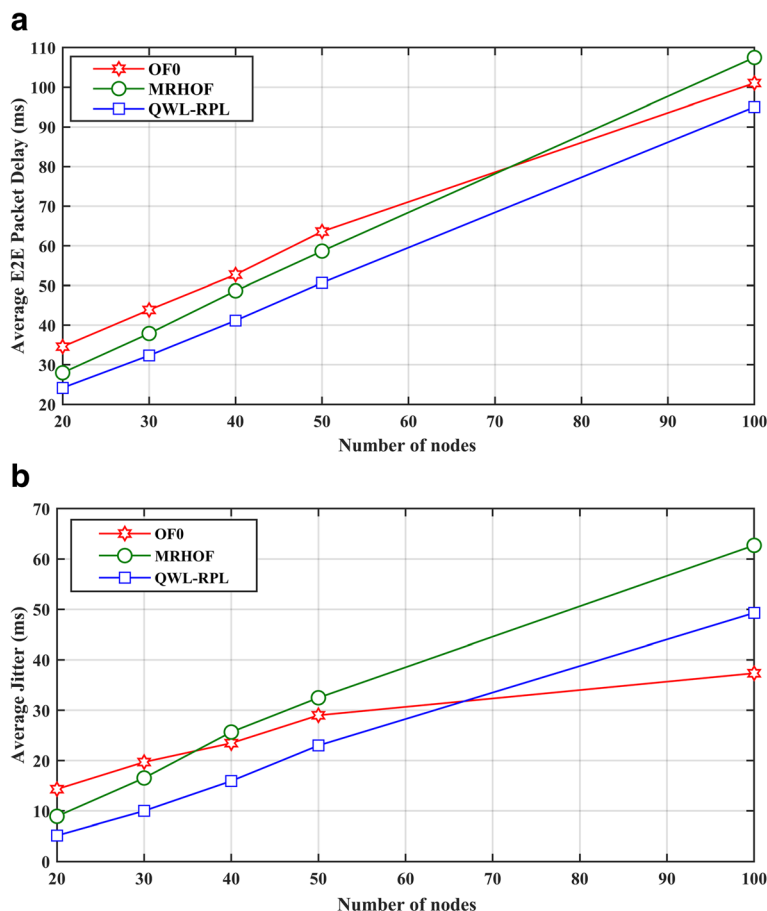
Table 4 Illustrating scenario I network imbalance with delivery ratio (%).

Objective functions	Network size (nodes)				
	20	30	40	50	100
	Number of nodes with delivery ratio < 10%				
OF0	0	3	4	10	56
MRHOF	0	0	0	1	3
QWL-RPL	0	0	0	0	0

Table 5 Illustrating scenario II network imbalance with delivery ratio (%).

Objective functions	Network size (nodes)				
	20	30	40	50	100
	Number of nodes with delivery ratio < 10%				
OF0	0	3	7	19	76
MRHOF	0	0	0	0	3
QWL-RPL	0	0	0	0	2





**Fig. 9** Simulation analysis for scenario I for the **a** average E2E delay (ms) and **b** average jitter versus number of nodes

utilizes IEEE 802.15.4 links for communication, carrier sense multiple-access (CSMA) for the MAC protocol, and ContikiMAC for asynchronous radio duty cycling with the default sleep interval of 125 ms. All nodes transmit their packets in a discrete time-slotted manner ( $t = 1, 2, 3...$ ). The maximum retransmission limit for the link layer is eight retransmission attempts. The CC2420 radio clear channel assessment (CCA) threshold (CC2420\_CONF\_CCA\_THRESH) is  $-45$  dBm.

The nodes are stationary and utilize the Zolertia Z1 mote platform [56]. Z1 mote provides new features such as maximum efficiency and robustness with low energy cost. It also provides a bigger ROM size (96 KB) as compared with other motes such as TMote Sky that provide ROM of size 48 KB. The Z1 devices also have one of the lowest payload sizes, i.e., 140 bytes. The TMote Sky payload size is 240 bytes. Thus, it is a good platform to fit typical sensor messages into a single IP packet of the uIP stack. The Z1 mote is ported to the Contiki OS (platform/z1) in a simulation environment. This mote also uses the MSP430F2617 low power

microcontroller, with 8 KB RAM and a 92 KB flash memory and runs the IEEE 802.15.4-compliant CC2420 transceiver. Each node transmits a packet size of 127 bytes and the buffer occupancy (Queue\_Conf\_Num) of each node is 4 packets. The maximum payload size supported by the uIP stack (UIP\_CONF\_BUFFER\_SIZE) is set to 140, which means any IP packet with size more than 140 will be dropped.

### 3.2 The ConikiRPL implementation and analysis with OF0 and MRHOF

This sub-section describes the experimental simulation scenarios with the default OF0 and MRHOF objective functions. The network is running with default objective functions to choose the best routing path. In the ConikiRPL implementation, choosing either OF0 or MRHOF requires the setting of various parameters. These parameter settings and their associated files are shown in Table 1.

For OF0, Cooja uses a rank with a minimum of 256 units (min\_hoprankinc) that allows a maximum of 255 hops, and for MRHOF, Cooja uses a rank (min\_

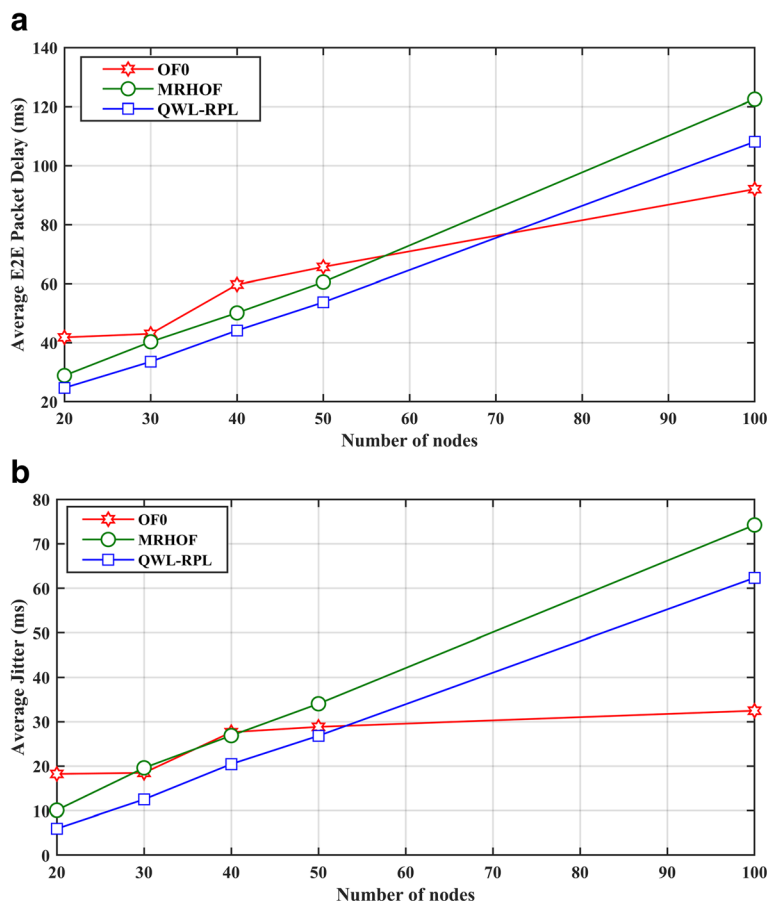


Fig. 10 Simulation analysis for scenario II for the a average E2E delay (ms) and b average jitter versus number of nodes

hoprankinc) with a minimum unit of 128. The ETX metric (parent\_link\_metric(p)) starts with a unit of 256 with a fixed-point divisor of 128 (LINK\_STATS\_ETX\_DEVISOR). The ETX value increases or decreases according to the MAC layer retransmission attempts. ETX is the expected number of transmissions required to transmit a packet successfully. ETX primarily reflects the packet loss by considering the quality of the wireless channel and packet collisions. In other words, a lower ETX means that the node is attempting to send a packet successfully in a few attempts and is therefore also consuming less energy. The ETX metric does not reflect the delay; if the packet is transmitted successfully, the delay metric is not affected.

ETX also depends on the distance between two nodes because the probabilities of successful transmission decrease with the increase of distance. ETX can also be estimated based on the RSSI value (guess\_etx\_from\_rssi). ETX can be reduced by increasing the transmission power of the nodes but this also increases the interference range in the wireless medium. ETX can be represented as follows:

$$ETX = \frac{1}{Df \times Dr} \tag{1}$$

where Df indicates the probability of packets being received by the neighboring node and Dr is the probability that the acknowledgment is received successfully. In the Contiki OS, ETX is based on the callbacks from the CSMA protocols which provides information on how many attempts were required in transmitting a packet. RPL uses an exponentially weighted moving average (EWMA) filter to smooth the ETX over time (EWMA\_ALPHA).

For a scenario of 20 nodes with a traffic generation rate of 1, 10, 30, 60 packets per minute (ppm), the results with OF0 and MRHOF with the ETX metric are depicted in Table 2. With an unbalanced traffic load, the PRR of OF0 is 78.84 percent and the overheads are 60472 units. The MRHOF increased the network efficiency in terms of the performance metrics mentioned in Table 2, but still produced 17007 overheads. In a scenario of many networks with an unbalanced traffic generation rate of each node, the performance would worsen further. The sensor node with better link quality forwards more packets and, in the

case of uneven data traffic, this results in significant load imbalance. The energy depletion rate will also be greater, thus causing holes in the network.

---

**Algorithm 1:** Workload measurement.

---

1. **Initialize:**
2. packet transmission (numtx)
3. get last transmission clock time
4. Set event timer of 10 clock seconds
5. Workload  $\leftarrow$  workload + numtx
6. **if** (event timer expired), **then**
7.     workload  $\leftarrow$  0
8.     reset event timer
9. **end**
10. Move to step 5

---

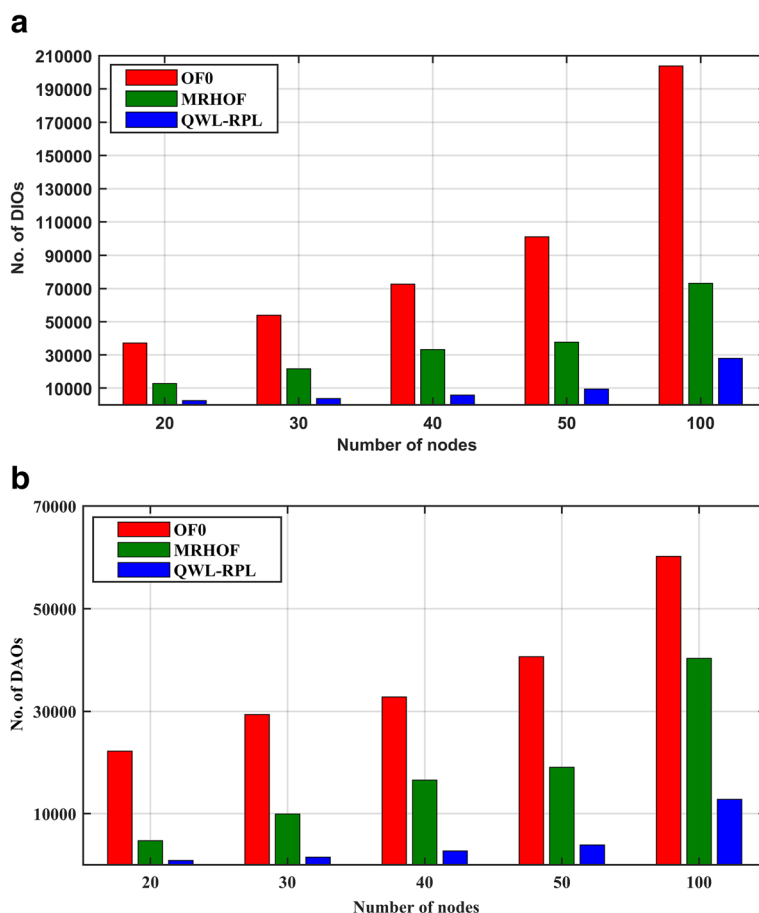
### 3.3 QWL-RPL algorithm implementation and analysis

The QWL-RPL algorithm considers the load imbalance problem during a heterogeneous traffic scenario. As we have observed in the previous sub-section, the OF0 and ETX-based MRHOF is not suitable for solving the load imbalance problem caused by traffic heterogeneity. Thus, using a subtree size in the routing table for load balancing might not be suitable in varying traffic load conditions. For this type of scenario, we are utilizing the periodic

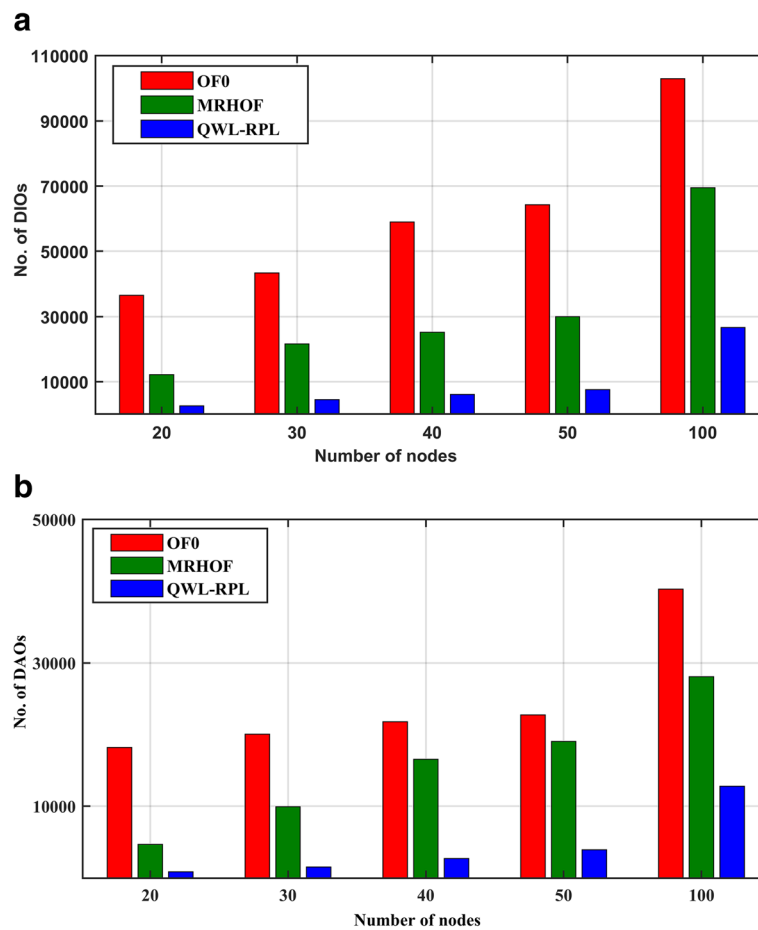
workload information along with the queue status of each node. The queue of each node can hold up to 4 packets at a time due to the smaller memory size of the Z1 mote. This amount of data is considerably smaller compared with the workload information. The workload of each node consists of the self-load and the load due to the subtree (the child nodes). The number of transmissions of each node every 10 s describes the workload information. In this work, the node workload is obtained based on the number of total packets it has transmitted in the last 10-s duration, which includes the packets it received from the subtree nodes and the number of the packets generated by the parent node itself. The workload calculation is described in Algorithm 1. The weighted queue information is utilized along with the workload parameter.

$$\text{Rank} = \text{Rank}(p) + \alpha Q_{\text{buf\_len}} + WL_{\text{MACtx}} \tag{2}$$

where  $\text{Rank}(p)$  is the parent node rank,  $Q_{\text{buf\_len}}$  is the number of packets in the packet queue buffer, and  $WL_{\text{MACtx}}$  is the number of transmitted packets at the MAC layer during the last time period. As explained previously,



**Fig. 11** Simulation analysis for scenario I for the **a** total number of DIO overheads and **b** total number of DAOs versus number of nodes



**Fig. 12** Simulation analysis for scenario II for the **a** total number of DIO overheads and **b** total number of DAOs versus number of nodes

the ETX-based node broadcasts probe packets at each interval to assess the link quality, and the receiving node re-broadcasts this probe packet, which further increases the network congestion as well as delay. To alleviate the large amount of overhead and increase the average delay, we did not utilize the ETX metric in the proposed mechanism. The  $\alpha$  is a positive integer that is used as a weighting factor in our experiment to enhance the impact of a small queue size for the routing decision. The value of the weighting factor  $\alpha$  is calculated empirically. According to the empirical results obtained in the simulation experiment, the optimal weighting factor is 90. At this value, the PRR is higher and the average delay is less. The average delay is the key parameter

in a heterogeneous network where some of the nodes might transmit voice/video packets. Figure 4 depicts the PRR in percentage and average delay against the varying weighting factor. The weighting factor is an important parameter that affects the network performance. Our empirical results show that, at  $\alpha = 90$ , the proposed protocol provides the highest PRR and lower average delay.

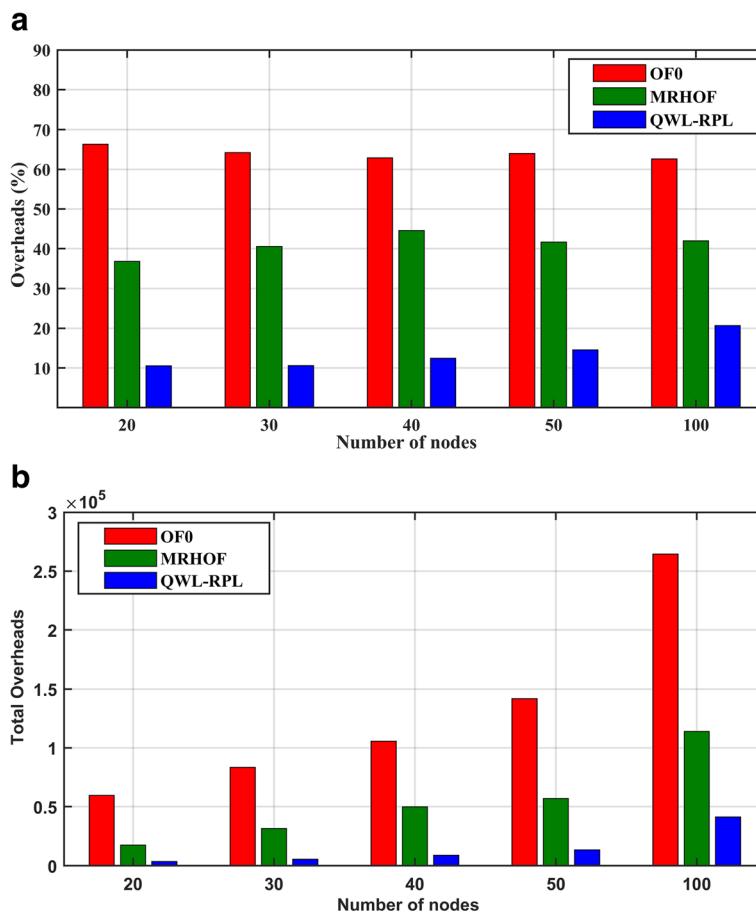
In the Contiki OS Cooja simulator, the metric container (MC) of ETX is defined while the MC of energy is also present as a stub. We can either develop our own MC or we can utilize the already available MC. In either of the cases, we need to update the rank information to the child nodes. The transmission of the DIO message follows the

**Table 6** Illustrating number of DIS messages for scenario I

Network size (nodes)	Number of DIS messages
20	59
30	96
40	141
50	185
100	439

**Table 7** Illustrating number of DIS messages for scenario II

Network size (nodes)	Number of DIS messages
20	62
30	91
40	124
50	156
100	384



**Fig. 13** Simulation analysis for scenario I for the **a** overheads (%) and **b** the total number of overheads versus number of nodes

trickle mechanism. The LLN nodes have limited computational capability and a limited energy source. Thus, keeping the transmission of overhead packets at a minimum is an essential part of RPL network. To achieve this goal, RPL utilizes a mechanism called the trickle timer [57]. The trickle timer is used to schedule the control messages and service discovery. The DIO packets are transmitted more frequently if the network is not stable or a new node enters the network to update the DODAG faster [58–60].

**Algorithm 2:** Rank calculation.

```

1.  if (node = root node), then
2.    root rank ← 128
3.  end
4.  if (parent = null), then
5.    rank ← max path cost * 128 (i.e., 100 * 128)
6.  end
7.  if (parent != null), then
8.    rank ← rank(p) + rank increase
9.    rank increase = α Qbuf_len + WLMACtx
10. end
11. if (rank (p) = 0), then
12.   rank ← base rank (128)
13. end
14. return MIN (base rank + rank increase)
    
```

After receiving the DIO message, each node generates the list of candidate parent nodes. During the selection procedure, the parent node with the minimum rank value is selected. The simulation results are presented in the next section. The rank calculation is shown in Algorithm 2. In RPL, the rank of the node is based on its parent rank and base rank. The default value of base rank is 128. If the node has no parent and the base rank is zero, then the rank is infinite. If the base rank is not zero, the rank is the base rank plus the rank increase. The rank increase is calculated according to the OF which is based on the queue and workload parameters in our case. The child node contains the list of candidate parent node ranks, compares the two or more parents, and returns the minimum ranked parent node.

#### 4 Simulation method and experiment

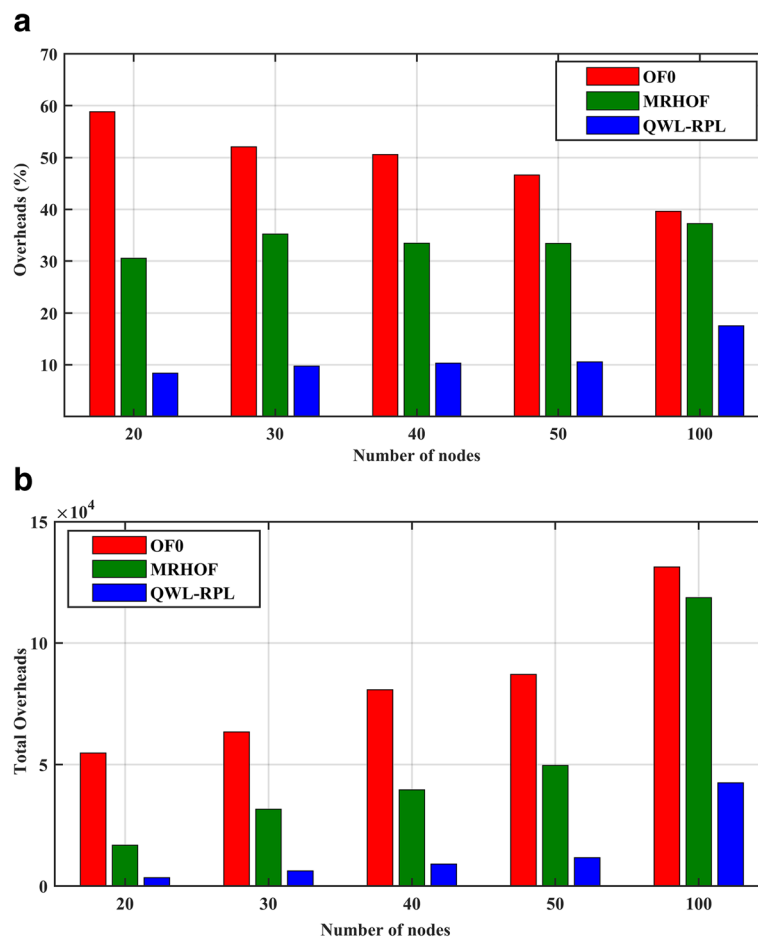
In this section, we perform the simulation study of the proposed mechanism. The Contiki OS is used to test RPL in the simulation environment. The performance of the proposed mechanism is evaluated in a network carrying heterogeneous traffic. The analysis is first

performed on a fixed heterogeneous traffic pattern; then the network is evaluated with a varying traffic pattern. Both types of traffic environments are compared with two other OFs, i.e., OF0 and MRHOF. The performance of these three OFs is evaluated in terms of PRR, packet loss ratio (PLR), average delay, jitter, number of control overheads, convergence time, and energy consumption. The performance is evaluated for five different network sizes. These network sizes consider the topology of 20, 30, 40, 50, and 100 nodes with one sink node in each network. All nodes send packets to the one sink node. The simulations are repeated a number of times. The simulation parameters are shown in Table 3.

Figure 5 shows the routing topology with a varying number of nodes. Each node is a Z1 Zolertia node, and the colors of nodes represent the different traffic patterns. The sending interval of the nodes varies with respect to the clock second; for example, one packet \* clock second means a node transmits one packet after every 1 s (i.e., 60 ppm); two packets \* clock second shows a node is transmitting one packet every 2 s (i.e., 30 ppm); six packets \* clock second describes a transmission rate

of one packet every 6 s (i.e., 10 ppm); similarly a pattern of 60 packets \* clock second defines a node transmission rate of one packet every 60 s (i.e., 1 ppm). In this scenario, the first simulation is performed with 20 clients (senders) nodes as depicted in Fig. 5a using OF0, MRHOF, and QWL-RPL. Similarly, network scenarios with 30 nodes, 40 nodes, 50 nodes, and 100 nodes are depicted in Fig. 5b, c, d, and e, respectively. There is one sink node (node one) and all the OFs use similar network configurations.

Figure 6 shows scenario II of the same network configuration except all the client's nodes are generating traffic randomly. The packet generation is based on a random number of 1–15 \* clock second/random 1–10. The random number seeds are generated with respect to clock time (clock\_time()) each iteration. For scenario II, Fig. 6a shows a topology with 20 nodes, and Fig. 6b, c, d, and e show topologies with 30, 40, 50, and 100 nodes, respectively, generating network traffic randomly. To calculate the energy consumption, we have utilized the Energest module in the Contiki OS.



**Fig. 14** Simulation analysis for scenario II for the **a** overheads (%) and **b** the total number of overheads versus number of nodes

### 5 Experimental results and analysis

This section provides an analysis of the results of the simulations. We used Python 3.7 to analyze the raw data obtained from the Cooja simulator. We studied the behavior of these protocols using two heterogeneous traffic scenarios and evaluated these protocols by varying the number of deployed sensor nodes. By increasing the number of nodes, more traffic is generated thus leading to the congestion. We deployed networks up to 100 nodes in size to study the default and proposed protocols under dense nodes and a heavy load environment.

PRR shows the percentage of the number of packets that are successfully received at the sink node relative to the number of packets sent from all the nodes to the sink node. PRR is calculated with the formula  $PRR = (\text{total packets received} / \text{total packets sent}) \times 100$ . Heterogeneous traffic networks are a common phenomenon in a number of IoT applications, such as industrial monitoring, health care [61] [62], or the smart home [63]. PRR can be better addressed with the proposed scheme in a heterogeneous load environment. Figure 7a shows the PRR of the OF0, MRHOF, and the QWL-RPL

protocol analyzed in a fixed heterogeneous traffic network scenario. Similarly, PLR (the packet loss ratio) is depicted in Fig. 7b. The PLR is calculated with the formula  $PLR = (100 - PRR)$ . PRR, on the other hand, shows the performance of the protocol in terms of the percentage of packets successfully delivered to the sink node. QWL and MRHOF develop more reliable networks as compared with OF0. The reason is that OF0 has no link reliability mechanism and the routes are selected based only on the hop counts. Although the ETX-based MRHOF applies a probing mechanism to measure the link quality, it still shows slightly lower PRR as compared with the QWL protocol. Figure 8a and b shows a simulation analysis of PRR and PLR for varying heterogeneous traffic networks (scenario II depicted in Fig. 6).

The QWL protocol congestion- and workload-aware mechanism causes the node to route traffic on the path with the lighter load to cause less packet loss. The RPL protocol does not have a mechanism to detect a load imbalance caused by the heterogeneous traffic load. The packets are backlogged at the node having a high packet generation by itself or by its child nodes in a short interval.

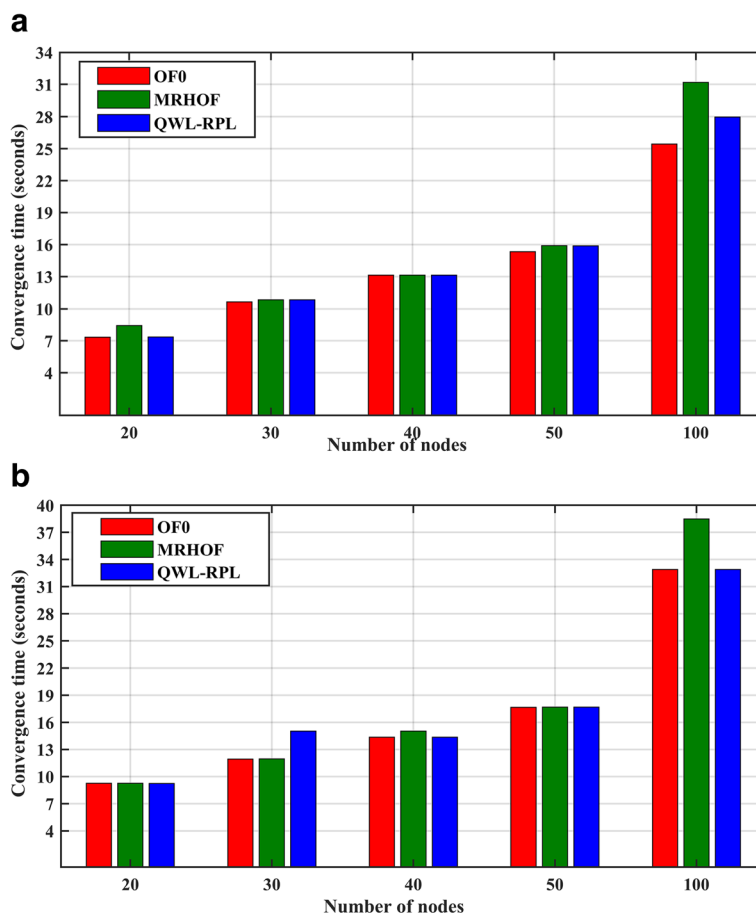


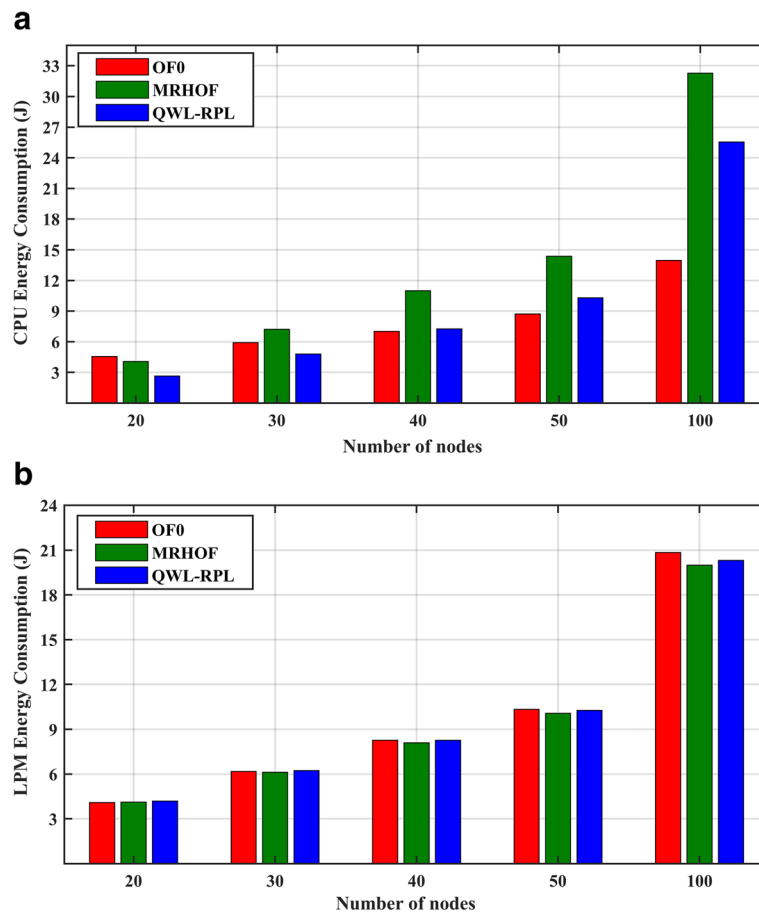
Fig. 15 Simulation analysis for the a scenario I convergence time and b scenario II convergence time versus number of nodes

This results in congestion and packet loss. During our experiment, we noticed that some of the nodes lost a large number of packets; for example, Tables 4 and 5 demonstrate the number of nodes having a packet delivery ratio of less than 10% in scenario I and scenario II, respectively. It can be seen that none of the nodes in the QWL-RPL network have less than 10% delivery ratio while OF0 has many nodes with a low ( $< 10\%$ ) delivery ratio. The MRHOF also contain such node(s) in 50- and 100-node network size. Similarly, we also noticed that in OF0 and MRHOF there are some nodes that were unable to transmit any packets, i.e., their delivery ratio is 0%. For example, with the 100-node scenario (Fig. 5e), we noticed all the packets of node number 101 are lost in MRHOF. Similarly, nodes 10, 27, 69, 71, 76, 79, 80, 85, 90, 95, 96, and 101 were not able to deliver any packets to the root node in the OF0 network. The delivery ratio of these nodes is 0%. We want to highlight that no node in the QWL network had a delivery ratio of less than 10%.

The average end-to-end packet delivery delay (E2E) represents the average delays of all the packets in the network. E2E delay is the time taken by a packet to travel from its

origin to the sink node. Figures 9a and 10a show the average E2E packet delay for all three protocols in scenario I and scenario II, respectively. The E2E delay in OF0 is greater but is reduced by increasing the number of nodes. The increased delay is because OF0 finds the path with the lowest hop count irrespective of congestion. In a large-size network utilizing OF0, some of the nodes are unable to deliver any packet as explained previously. We measured the delay of packets transmitted successfully to the sink node. MRHOF shows more delay due to the continuous probing mechanism, whereby the path with good link quality is found using probe packets. Some of the nodes or routing paths may be congested or overloaded, which forces packets to be queued. Thus, both link-level and load-level congestion result in more delays. QWL utilizes the queue and workload information simultaneously, which helps the node to find a less congested path efficiently.

We also evaluated the quality of service performance metrics in terms of jitter. Jitter is a delay that varies over time, i.e., the variation in delay. Delay represents the E2E delay of packets; whereas, jitter is the variation of packet arrival times and is calculated as the delay between two



**Fig. 16** Simulation analysis for scenario I for **a** CPU energy consumption (J) and **b** LPM energy consumption versus number of nodes



consecutive packets. Average network jitter, calculated as  $Jitter = (total\ Jitter/number\ of\ nodes)$ , is an important parameter in time-critical applications where each packet must reach its destination with a delay constraint relative to the previous packet delay. The video or image sequential transmission requires a strict timing regularity in delivering the stream of packets successfully to maintain a smooth reception of packets at the destination. The results of jitter observed in the simulation experiment are shown in Figs. 9b and 10b respectively. The jitter occurs due to the congestion in the network. The jitter was reduced in OF0 when the network size reached 100 nodes. As explained in the E2E delay, some of the nodes in OF0 could not transmit any packets. The MRHOF has more average delay due to congestion and improper load balancing and, thus, causes more jitter.

The sensor node has a very limited capacity in terms of energy, memory, and computational capability. Along with the data packets, the control messages are exchanged in the network. The RPL incurs control overheads (DIO, DAO, and DIS) during the DODAG construction. After the construction of the network, the

nodes exchange the control messages to maintain the network using the trickle timer algorithm. These overheads lead to higher energy consumption, congestion, and collisions. In a stable network, fewer overheads are exchanged to maintain the connectivity. Based on the network condition, each node transmits DIO messages using the trickle timer. The frequency of DIO messages depends on network stability. Figure 11a and b depicts the total number of DIO overheads in scenario I and Fig. 12a and b shows the total number of DIOs and DAOs transmitted in scenario II. The nodes use DIS messages to probe the neighboring nodes for nearby DODAGs. The network created using all three mentioned OFs has the same number of DIS messages. The total number of DIS messages for each network size in scenario I and scenario II are depicted in Tables 6 and 7, respectively. The DIO messages are also related to PRR, i.e., the network with poor PRR incurs more DIO messages. The child nodes transmit their destination information towards the sink node via DAO messages. DAO messages also contribute to a large number of overheads. The DAO messages are transmitted to the sink when

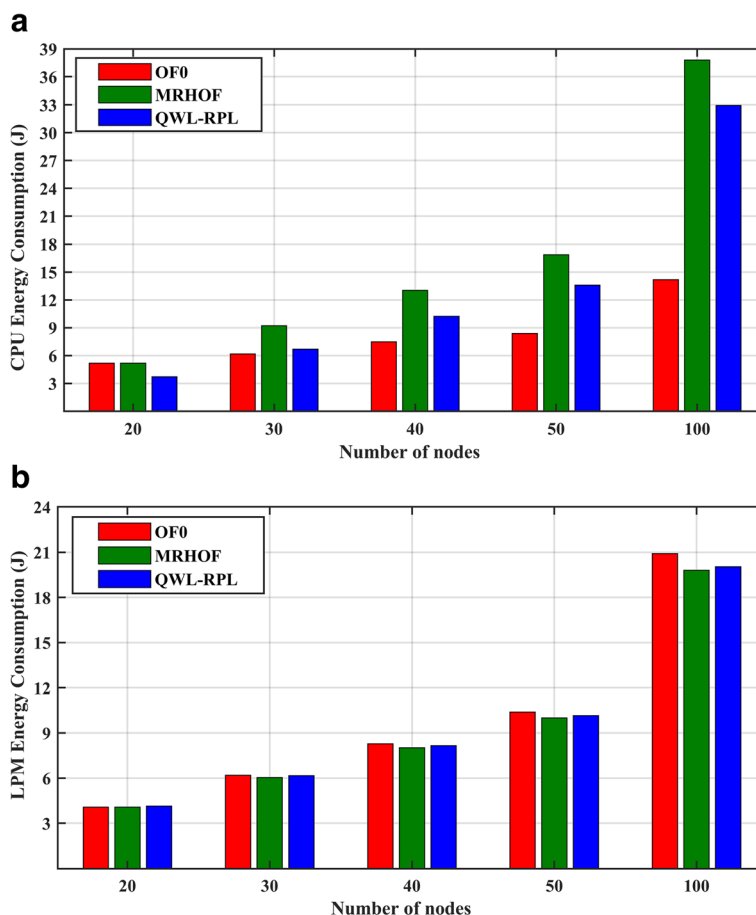
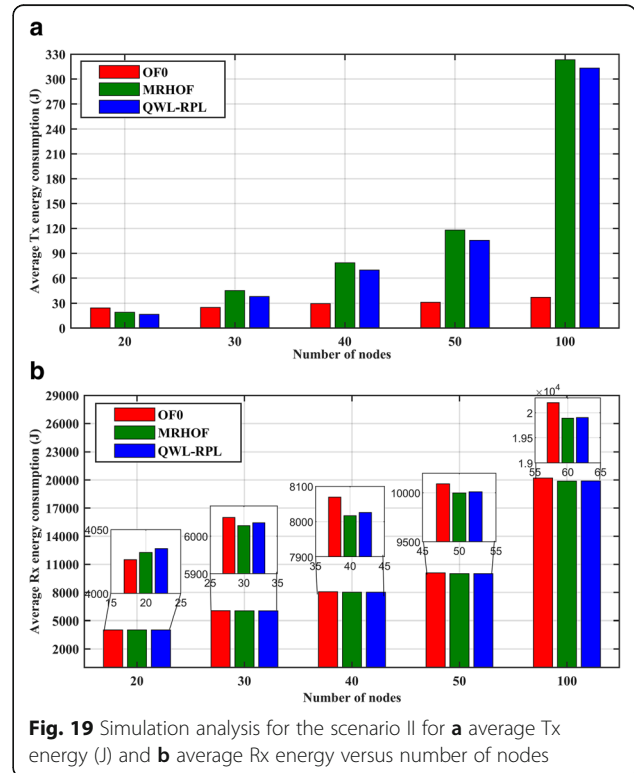
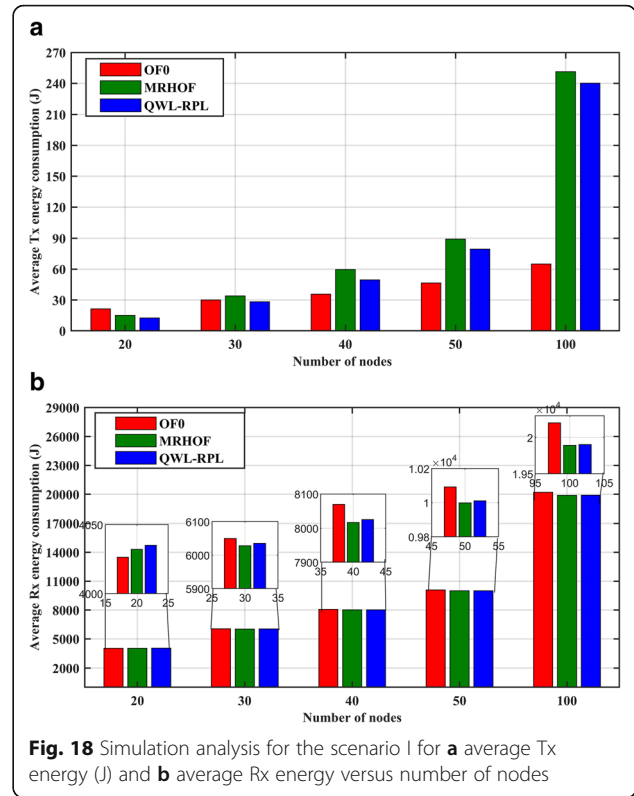
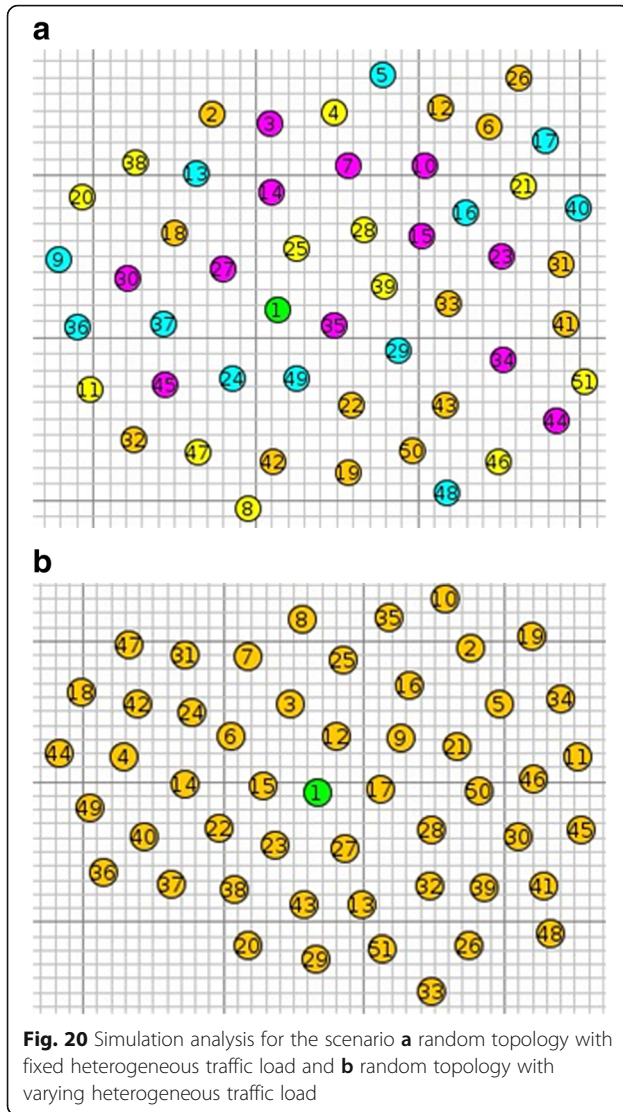


Fig. 17 Simulation analysis for scenario II for a CPU energy consumption (J) and b LPM energy consumption versus number of nodes

the upward route is changed. DIO and DAO messages maintain the routing topology by transmitting routing information periodically. Their transmission frequency decreases when the network is stable, and vice versa. These results reflect that a large amount of energy is wasted for routing signaling.

During the network initiation phase, all the packets in the network are control messages. Then, after network construction is completed, the nodes begin packet transmission. The overhead percentage compared with the total traffic in the network is shown in Figs. 13a and 14a for scenario I and scenario II, respectively. In Fig. 13a, the percentage of overheads in OF0 is more than 65% in all five network size scenarios. This means that a large percentage of limited battery power is wasted in transmitting and receiving the control overheads. This is because the OF0 cannot resolve the congestion and load-balancing issue, resulting in a highly unstable network. Similarly, MRHOF has a high percentage of overhead packets, as compared with QWL-RPL, due to the same congestion, retransmission, and load-balancing problem. The percentage of overheads in the QWL-RPL network is much less than that of OF0 and MRHOF; even for a large and dense network of 100 nodes, the total percentage of overheads is approximately 20.68% compared with 62.61% in OF0 and 41.94% in MRHOF. QWL-RPL is able to minimize the routing overheads by considering the node queue and workload status to route the packets through less congested and less overloaded paths. Similarly, the total number of overheads is also depicted in Figs. 13b and 14b for scenarios I and II, respectively. The total number of overheads is the sum of the total numbers of DIOs, DAOs, and DIS messages. The two dominant control overheads contributing to the large percentage of overheads are the DIOs and DAOs. The DAO messages are responsible for maintaining the downward routes. DAO overheads are different from DIO as each DIO message is broadcast while DAO messages are forward up to the sink node. Due to high PLR in OF0, more packets are retransmitted. The number of hops increases with the increase of network size and more control packets are transmitted to update the routing table entries. The number of DIO messages in MRHOF is higher than that of QWL, due to fluctuating link conditions, requiring more control packets to estimate the link qualities. The collisions due to probe packets are also one of the causes of irregular ETX. We can see the significant improvement that results from reducing the total number of control overheads in the network. The graphs representing QWL-RPL reflect that the network is consistent, stable, less congested, and consumes less energy resources. Thus, the QWL-RPL incurs fewer retransmissions and congestion.





We also evaluated the network in terms of convergence time. Convergence time is defined as the amount of time needed for all the nodes to join the network. To determine the convergence time, we require the time at which the first node joins the DAG and the time at which last node joins the DAG, measured with the formula convergence time = last node join time – first node join time. From this formula, it is obvious that the

network convergence time increases as the node density increases. This is also shown in Fig. 15a and b for scenario I and scenario II. The slight variation in convergence time is due to the latency or jitter and the time consumed by packet processing. When the node density increases significantly, the MRHOF shows higher convergence time. This difference is caused by the longer processing time in estimating the link quality. The ETX calculation in MRHOF requires a slightly longer time and becomes significant when network size increases to 100 nodes.

The energy is the scarcest resource in the sensor network and communications are where most of the energy is consumed. Thus, we provide a detailed analysis of energy consumption. Each node uses the Z1 mote platform as described previously. The approximate current consumption of the Z1 mote is categorized as follows. When the radio is OFF, the microcontroller is in an idle state which is referred to as the low-power mode (LPM). During the LPM, the current consumption is 20  $\mu$ A. The condition when the radio is off and the microcontroller is in the ON state is called the CPU idle state (CPU). During CPU idle state, the current consumption is 42.6  $\mu$ A. The current consumption is comparatively highest during transmission and reception periods. The transmission current (Tx) is 17.4 mA and reception current (Rx) is 18.8 mA. These parameters are according to the Z1 mote standard specifications [64]. The tick-per-second (RTIMER\_SECOND) value for the Z1 mote is 32786. RTIMER\_SECOND is used to convert the ticks into seconds. The voltage value is 3 V for all four stages. We calculated the cumulative energy consumption of all four stages for each network scenario. Note that the listening period energy consumption also includes the total reception energy consumption. The following formulas are used to measure the energy consumption in each state:

$$\text{LPM} = \text{LPM} \times 0.020 \times 3/32,768 \quad (3)$$

$$\text{CPU} = \text{CPU} \times 0.426 \times 3/32,768 \quad (4)$$

$$\text{Tx} = \text{Tx} \times 17.4 \times 3/32,768 \quad (5)$$

$$\text{Rx} = \text{Rx} \times 18.8 \times 3/32,768 \quad (6)$$

The LPM, CPU, Tx, and Rx values in ticks are obtained from Energest function (energest\_flush()) in the

**Table 8** Analysis of random topology with fixed heterogeneous traffic load

No. of nodes	OF	Total pkts sent	Total pkts received	Total packets lost	PRR %	Packet	Avg delay (ms)	Jitter (ms)	Total overheads %	Network energy consumption (mJ)
50 nodes	OF0	79,837	18445	61,392	23.1	76.9	49.12	18.94	50.18	10,102,852.89
50 nodes	MRHOF	79,833	50397	29,436	63.13	36.87	44.11	20.65	29.55	10,114,366.48
50 nodes	QWL-RPL	79,842	52509	27,333	65.77	34.23	40.25	16.27	8.04	10,109,797.83

**Table 9** Analysis of random topology with varying heterogeneous traffic load

No. of nodes	OF	Total pkts sent	Total pkts received	Total packets lost	PRR %	Packet lost %	Avg delay (ms)	Jitter (ms)	Total overheads %	Network energy consumption (mJ)
50 nodes	OF0	99,826	19,978	79,848	20.01	79.99	50.66	20.72	39.55	10,142,374.59
50 nodes	MRHOF	99,436	58,058	41,378	58.39	41.61	46.0	23.12	26.85	10,141,644.17
50 nodes	QWL-RPL	99,606	60,078	39,528	60.32	39.68	42.45	20.0	7.39	10,140,776.65

Contiki OS. A performance comparison between OF0, MRHOF, and QWL-RPL in terms of energy consumption is shown in Figs. 16, 17, 18, 19. Most of the energy is consumed during the reception and transmission times. The energy consumption of the network increases when more packets are delivered successfully. The CPU is utilized more in MRHOF and, thus, energy consumption is highest (Fig. 16a). The main reason for the higher energy consumption is the calculation of ETX, which is a complex procedure requiring the periodic probing mechanism. During low power mode, the sensor saves energy by turning off the radio. The LPM energy consumption is very small and is similar in each protocol (Fig. 16b). The similar pattern of CPU and LPM energy consumption is observed in scenario II (Fig. 17a and b). The greatest energy consumption occurs during Tx and Rx periods. Even though the total amount of overhead in OF0 is the highest, the total Tx energy consumption is greater in MRHOF, due to the poor delivery ratio of OF0. Most of the data packets are dropped in OF0; thus, the nodes consume less energy in transmission and it is evident that the energy consumption for data packets is greater than for control packets due to their relative packet sizes.

As the network size increases, the Tx energy consumption of QWL-RPL also increases (Figs. 18a and 19a) because of the differences between PRR and PLR. The OF0 has a very high percentage of overhead as compared with QWL-RPL. The percentage difference between data packets and overheads is described in Figs. 15 and 16. For example, for the case of 100 nodes in scenario I, the percentage of overheads transmitted in OF0 is 62.61% while QWL-RPL transmitted 20.68% overheads. Similarly, the PLR in OF0 is 87.4% and PLR in QWL-RPL is 55.72%. The similar result pattern is observed for scenario II.

From these results, we can understand the OF0 is consuming most of the energy in overhead transmissions while most of the energy spent in QWL-RPL is during sensor packet transmission. The MRHOF transmitted 41.94% overheads. Similarly, in the case of Rx energy consumption. The difference is highlighted by the magnifier in Figs. 18b and 19b. The simulation runs for 3600 s. The sensor nodes are meant to run for months or even for years in remote areas. The energy differences are significant for a network designed to run for a long period of time.

For proof of concept, we also simulated the proposed mechanism with the random topology of 50 nodes as shown in Fig. 20a and b. The performance assessment of the random topologies with fixed heterogeneous traffic load and varying traffic load is shown in Tables 8 and 9. From these results, we can see better equilibrium performance assessment results of QWL-RPL.

Similar to Z1 mote platform, we also evaluated the network with Tmote Sky [65] to provide a further representation of the proposed solution. The network evaluation using Tmote Sky is shown in Table 10. Considering the metrics used in the simulation, the advantages of the proposed solution compared with the standardized protocol is summarized in Table 11. As OF0 has no link reliability mechanism, the PRR is significantly low when using OF0 compared with MRHOF and proposed QWL-RPL. Similarly, the average delay is higher in OF0 and MRHOF compared with QWL-RPL. OF0 route the packets through the shortest path irrespective of the congestion. Similarly, MRHOF uses continuously probing mechanism that generates more control packets and thus causes congestion. The similar performance trend is observed for jitter and a total number of overheads.

**Table 10** Analysis of Tmote Sky mote

No. of nodes	OF	Total pkts sent	Total pkts received	Total packets lost	PRR %	Packet lost %	Avg delay (ms)	Jitter (ms)	Total overheads %
30 nodes	OF0	57,360	17,358	40,002	30.26	69.74	70.95	31.78	43.73
30 nodes	MRHOF	56,933	46,383	10,550	81.47	18.53	52.08	23.19	26.43
30 nodes	QWL-RPL	57,094	47,877	9217	83.86	16.14	44.89	15.63	6.46

**Table 11** Proposed protocol comparison with standardized protocol

OF	PRR %	Packet lost %	Avg delay (ms)	Jitter (ms)	Total overheads %
OF0	Low	High	High	High	High
MRHOF	Medium	Medium	Medium	Medium	Medium
QWL-RPL	High	Low	Low	Low	Low

## 6 Conclusion and future work

The RPL routing protocol is designed to support the LLNs characteristics. The LLN applications include diverse application scenarios from basic temperature measurements to the high-volume multimedia services that require efficient communication support. It is necessary to study the mechanism of RPL within such varying traffic application scenarios to develop improved features of RPL for the resource-constrained nodes. In this study, we addressed the load imbalance problem in the heterogeneous traffic scenario. We proposed a new objective function for the RPL protocol that alleviates the congestion occurrence owing to the imbalanced traffic load. We experimented with varying traffic loads and investigated whether traffic variations at each interval affected the overall performance of the proposed protocol. We studied two cases where the first case is based on fixed heterogeneous traffic pattern, while the second case discusses the varying heterogeneous traffic load. Using simulation analysis, we discussed that the default objective functions, i.e., OF0 and MRHOF, cannot efficiently support the traffic heterogeneity. To support such scenarios, we have proposed a queue- and workload-based solution (QWL-RPL) that utilizes the weighted queue and periodic workload information to distribute the load efficiently. The proposed mechanism aims to balance the traffic distribution as well as to minimize a large number of control overheads, thereby increasing the PRR, reducing the E2E packet delay, and reducing jitter. Overall, during an imbalanced heterogeneous traffic situation, QWL makes load- and congestion-aware routing decisions. We conducted extensive simulations in the Contiki OS to verify the protocol performance. In all the network sizes examined, QWL-RPL shows better PRR while OF0 performs the worst. Similarly, in terms of E2E delay, jitter, amount of overhead, and energy consumption, QWL-RPL achieves better and more reliable performance.

The RPL routing procedure contains an exchange of expensive control packets. We can utilize a machine learning algorithm to provide the routing protocol with a self-learning, self-adaptive, and low-complexity routing model. Thus, an interesting future direction is to shift the need from rule-based routing to learning-based

computing and estimation in the sensor network. We also plan to implement the proposed protocol in a real testbed scenario.

### Abbreviations

6LoWPAN: IPv6 low-power wireless personal area network; DAO: Destination advertisement objects; DIO: DODAG information object; DIS: DODAG information solicitations; DODAG: Destination-oriented directed acyclic graph; E2E: End-to-end delay; ETX: Expected transmission count; ICMPv6: Internet control message protocol for IPv6; IETF: Internet engineering task force; IoT: Internet of Things; LLN: Low-Power and Lossy Networks; LPM: Low power mode; MAC: Medium access control; MRHOF: Minimum rank with hysteresis objective function; OF: Objective function; OF0: Objective function zero; PLR: Packets loss ratio; PRR: Packets reception ratio; QWL-RPL: Queue utilization and link workload based RPL; RDC: Radio duty cycling; RPL: Routing protocol for Low-Power and Lossy Networks; Rx: Reception current; Tx: Transmission current; Z1: Zolertia 1

### Acknowledgements

This research was supported in part by the Brain Korea 21 Plus Program (No. 22A20130012814) funded by the National Research Foundation of Korea (NRF), in part by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Centre) support program (IITP-2019-2016-0-00313) supervised by the IITP (Institute for Information & communications Technology Planning & Evaluation), and in part by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (2018R1D1A1A09082266).

### Authors' contributions

AM and YBZ conceived the main idea, designed the algorithm, and proposed the routing framework for heterogeneous traffic network. YBZ contributed equally and should be considered as co-first author. AM, YBZ, and Z performed the implementation of the proposition in the Contiki Cooja simulator. SWK contributed to the structuring, reviewing, and finalizing of the manuscript. All authors read and approved the final manuscript.

### Funding

The funding source is the same as described in the acknowledgments.

### Competing interests

The authors declare that they have no competing interests.

Received: 4 September 2019 Accepted: 2 January 2020

Published online: 20 January 2020

### References

1. A. Gubbi, R. Buyya, S. Marusic and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645–1660, Sep. 2013.
2. F. Al-Turjman, "Smart-cities Medium Access for Smart Mobility Applications in IoT," *Wiley Transactions on Emerging Telecommunications Technologies*, Aug. 2019. DOI. 10.1002/ett.3723.
3. M. A. Naeem, R. Ali, B.-S. Kim, S. A. Nor, S. Hassan, "A periodic caching strategy solution for the smart city in information-centric Internet of Things", *Sustainability*, vol. 10, no. 7, pp. 2576, 2018.
4. F. Al-Turjman, "A Rational Data Delivery Framework for Disaster-inspired Internet of Nano-Things (IoNT) in Practice", *Springer Cluster Computing*, vol. 22, no. 1, pp. 1751–1763, Nov. 2017.

5. I. Yaqoob et al., "Internet of Things architecture: Recent advances, taxonomy, requirements, and open challenges," *IEEE Wireless Communications*, vol. 24, no. 3, pp. 10–16, Jun. 2017.
6. A. Musaddiq, Y. B. Zikria, O. Hamm, H. Yu, A. K. Bashir, and S. W. Kim, "A survey on resource management in IoT operating systems," *IEEE Access*, vol. 6, pp. 8459–8482, Feb. 2018.
7. N. A. Pantazis, S. A. Nikolidakis and D. D. Vergados, "Energy-efficient routing protocols in wireless sensor networks: A survey," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 2, pp. 551–591, Second Quarter. 2013.
8. Tymo source code repository. Tymo: DYMO implementation for TinyOS, December 2007. (Accessed on 20/11/2019). [Online] Available: <http://tymo.sourceforge.net>.
9. K. Kim, S. Yoo, J. Park, S.D. Park, J. Lee, Hierarchical Routing over 6LoWPAN (HiLow), IETF: Internet Draft: draft-deniel-6lowpan- hlow-hierarchical-routing-00.txt, vol. 38, December 2005.
10. K. Kim, S. Daniel Park, G. Montenegro, S. Yoo, N. Kushalnagar, 6LoWPAN Ad Hoc On-Demand Distance Vector Routing (LOAD), IETF Internet Draft: draft-daniel-6lowpan-load-adhoc-routing-03, 2007.
11. K. Kim, S. Park, I. Chakeres, C. Perkins, Dynamic MANET On-demand for 6LoWPAN (DYMO-low) Routing, Internet Draft: draft- montenegro-6lowpan-dymo-low-routing-03, June 2007.
12. O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, P. Levis, Collection tree protocol, in *Proc. 7th ACM Conference on Embedded Networked Sensor Systems (SenSys)*, Berkeley, CA, USA, Nov. 2009, pp. 1–14.
13. A. Tavakoli, HYDRO: A Hybrid Routing Protocol for Lossy and Low Power Networks, IETF Internet Draft: draft-tavakoli-hydro-01, Sept 2009.
14. RPL: IPv6 routing protocol for low-power and lossy networks. (Accessed on 20/11/2019). [Online] Available: <https://tools.ietf.org/html/rfc6550>.
15. O. Gaddour, A. Koubâa, "RPL in a nutshell: A survey", *Computer Networks*, vol. 56, no. 14, pp. 3163–3178, Sep. 2012.
16. L. D. Xu, W. He and S. Li, "Internet of Things in industries: A survey," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 4, pp. 2233–2243, Nov. 2014.
17. D. Miorandi, S. Sicari, F. De Pellegrini, and I. Chlamtac, "Internet of things: Vision, applications and research challenges," *Ad Hoc Network*, vol. 10, no. 7, pp. 1497–1516, Sep. 2012.
18. Cisco, "Connected Grid Networks for Smart Grid - Field Area Network". (Accessed on 20/11/2019). [Online] Available: <http://www.cisco.com/c/en/us/solutions/industries/energy/external-utilities-smart-grid/field-area-network.html>.
19. P. Thubert, "Objective Function Zero for RPL," RFC 6552, vol. 33, pp. 3–8, 2012.
20. O. Gnawali and P. Levis, "The Minimum Rank with Hysteresis Objective Function", RFC 6719 (Proposed Standard), Internet Engineering Task Force, Sep. 2012. (Accessed on 20/11/2019). [Online] Available: <http://www.ietf.org/rfc/rfc6719.txt>.
21. P. O. Kamgueu, E. Nataf, T. D. Ndié, O. Festor, "Energy-Based Routing Metric for RPL", *INRIA*, Roquecourbe, France, 2013.
22. H.-S. Kim, H. Kim, J. Paek, and S. Bahk, "Load balancing under heavy traffic in rpl routing protocol for low power and lossy networks," *IEEE Transaction on Mobile Computing*, vol. 16, no. 4, Apr. 2017.
23. M. Palattella et al., "Standardized protocol stack for the Internet of (important) Things," *IEEE Communication Surveys Tutorials*, vol. 15, no. 3, pp. 1389–1406, Third Quarter 2013.
24. Contiki netstack. (Accessed on 20/11/2019). [Online] Available: [http://anrg.usc.edu/contiki/index.php/Network\\_Stack](http://anrg.usc.edu/contiki/index.php/Network_Stack).
25. R. Ali, N. Shahin, Y. B. Zikria, B. Kim, and S. W. Kim, "Deep reinforcement learning paradigm for performance optimization of channel observation-based MAC protocols in dense WLANs," *IEEE Access*, vol. 7, no. 1, pp. 3500–3511, Jan. 2019.
26. A. Dunkels, "The ContikiMAC Radio Duty Cycling Protocol," Technical Report, SICS, 2011.
27. M. Buettner, G. V. Yee, E. Anderson, R. Han, "X-MAC: A Short Preamble MAC Protocol for Duty-cycled Wireless Sensor Networks", *Proc. 4th International Conference on Embedded Networked Sensor Systems*, Boulder, CO, USA, pp. 307–320, Nov. 2006.
28. Contiki: The Open Source Operating System for the Internet of Things. (Accessed on 20/11/2019). [Online] Available: <http://www.contiki-os.org/>.
29. K. Bhandari, A. S. M. S. Hosen, G. H. Cho, "CoAR: Congestion-Aware Routing Protocol for Low Power and Lossy Networks for IoT Applications". *Sensors*, vol. 18, no. 11, 3838, Nov. 2018.
30. C. L. Hwang, K. Yoon. "Methods for Multiple Attribute Decision Making", *Springer*: Berlin/Heidelberg, Germany, 1981; pp. 58–191.
31. W. Tang, X. Ma, J. Huang, J. Wei, "Toward improved RPL: A congestion avoidance multipath routing protocol with time factor for wireless sensor networks", *Hindawi Journal of Sensors*, vol. 2016, Jun. 2016.
32. S. Taghizadeh, H. Bobarshad and H. Elbiaze, "CLRPL: Context-Aware and Load Balancing RPL for IoT Networks Under Heavy and Highly Dynamic Load," *IEEE Access*, vol. 6, pp. 23277–23291, Apr. 2018.
33. P. Di Marco, C. Fischione, G. Athanasiou and P. Mekikis, "MAC-aware routing metrics for low power and lossy networks," *2013 Proceedings IEEE INFOCOM*, Turin, Italy, 2013, pp. 13–14.
34. P. Sanmartin, A. Rojas, L. Fernandez, K. Avila, D. Jabba, S. Valle, "Sigma Routing Metric for RPL Protocol", *Sensors*, vol. 18, no. 4, pp. 1277, Apr. 2018.
35. S. Yang, Y. Baek, J. Kim, K. Cho, K. Han, "A routing metric for load balance in wireless mesh networks". In *Proc. of the International Conference on Advanced Communication Technology (ICACT)*, Phoenix Park, Korea, 15–18 Feb. 2009, pp. 1560–1565.
36. T. B. Oliveira, P. H. Gomes, D. G. Gomes, B. Krishnamachari, "ALABAMO: A LoAd BALancing MOdel for RPL", in *Proc. Brazilian symposium on computer networks and distributed systems (SBRC)*, pp. 644–657, Jun. 2016.
37. Xinxin Liu, Jianlin Guo, G. Bhatti, P. Orlik and K. Parsons, "Load balanced routing for low power and lossy networks," *IEEE Wireless Communications and Networking Conference (WCNC)*, Shanghai, China, 2013, pp. 2238–2243.
38. O. Banimelhem, S. Khasawneh, "GMCAR: Grid-based multipath with congestion avoidance routing protocol in wireless sensor networks", *Ad Hoc Network*, vol. 10, no. 7, pp. 1346–1361, Sep. 2012.
39. M. Conti, E. Gregori, G. Maselli, "Reliable and Efficient Forwarding in Ad Hoc Networks", *Ad Hoc Networks*, vol. 4, pp. 398–415, 2006.
40. L. Reddy, S. Raghavan, "SMORT: Scalable multipath on-demand routing for mobile ad hoc networks", *Ad Hoc Networks*, vol. 5, no. 2, pp. 162–188, Mar. 2007.
41. O. Iova, F. Theoleyre and T. Noel, "Improving the network lifetime with energy-balancing routing: Application to RPL," *7th IFIP Wireless and Mobile Networking Conference (WMNC)*, Vilamoura, Portugal, 2014, pp. 1–8.
42. P. Kamgueu et al, "Energy-Based Routing Metric for RPL," Research Report RR-8208, INRIA, 2013.
43. M. Zhao, G. G. M. N. Ali, R. Lu and P. H. J. Chong, "An Energy-Efficient and Self-Reasoning-Based RPL for Low-Power and Lossy Networks," *IEEE 84th Vehicular Technology Conference (VTC-Fall)*, Montreal, QC, Canada, 2016, pp. 1–6.
44. N. Sousa, J. V. V. Sobral, J. J. P. C. Rodrigues, R. A. L. Rabêlo and P. Solic, "ERAOF: A new RPL protocol objective function for Internet of Things applications," *2nd International Multidisciplinary Conference on Computer and Energy Science (SpliTech)*, Split, Croatia, 2017, pp. 1–5.
45. A. Barbato, M. Barrano, A. Capone, N. Figiani, "Resource oriented and energy efficient routing protocol for IPv6 wireless sensor networks," In *Proc. of the IEEE Online Conference on Green Communications*, Piscataway, NJ, USA, 29–31 Oct. 2013.
46. Y. Chen, J. Chanet, K. Hou, H. Shi, G. Sousa, "A Scalable Context-Aware Objective Function (SCAOF) of Routing Protocol for Agricultural Low-Power and Lossy Networks (RPAL)". *Sensors*, 2015, 15, 19507–19540.
47. O. Gaddour, K. Anis, B. Nauha, and A. Mohamed, "OF-FL: QoS-aware fuzzy logic objective function for the RPL routing protocol," In *Proc. International Symposium on Modelling and Optimization in Mobile, Ad Hoc, and Wireless Networks*, pp. 365–372, 2014.
48. P. O. Kamgueu, N. Emmanuel, and N. D. Thomas, "On design and deployment of fuzzy- based metric for routing in low-power and lossy networks", In *Proc. International conf. on Local Computer Networks Conference Works*, pp. 789–795, 2015.
49. W. Khallef, M. Molnar, A. Benslimane and S. Durand, "Multiple constrained QoS routing with RPL," *IEEE International Conference on Communications (ICC)*, Paris, France, 2017, pp. 1–6.
50. H. Yoo, M. Shim, D. Kim and K. H. Kim, "GLOBAL: A Gradient-based routing protocol for load-balancing in large-scale wireless sensor networks with multiple sinks," *The IEEE symposium on Computers and Communications*, Riccione, Italy, 2010, pp. 556–562.
51. E. Rezaei, "Energy efficient rpl routing protocol in smart buildings," Jun. 2014.
52. A. Barbato, M. Barrano, A. Capone, and N. Figiani, "Resource oriented and energy efficient routing protocol for ipv6 wireless sensor networks," in *Proc. IEEE GreenCom*, 2013, pp. 163–168.
53. A. Brachman, "RPL objective function impact on LLNs topology and performance," in *Proc. Internet of Things, Smart Spaces, and Next Generation Networking*, S. Balandin, S. Andreev, and Y. Koucheryavy, eds. Berlin, Germany: Springer, 2013, vol. 8121, pp. 340–351.
54. M. Z. Hasan, F. Al-Turjman, "SWARM-based data delivery in Social Internet of Things", *Elsevier Future Generation Computer Systems*, vol. 92, no. 1, pp. 821–836, Mar. 2019.

55. M. D. Silva, E. L. F. Senne, N. L. Vijaykumar, "An optimization model to minimize the expected end-to-end transmission time in wireless mesh networks," *Pesquisa Operacional*, 2017, 37, 209–227.
56. Zolertia Z1 mote. (Accessed on 05/13/2019). [Online] Available: <https://github.com/Zolertia/Resources/wiki/The-Z1-mote>.
57. P. Levis, N. Patel, D. Culler, and S. Shenker, "Trickle: A self-regulating algorithm for code propagation and maintenance in wireless sensor networks," in *Proc. 1st USENIX/ACM Symp. NSDI*, 2004, pp. 15–28.
58. P. Levis, T. Clausen, J. Hui, O. Gnawali, and J. Ko, "The Trickle algorithm," RFC 6206, Internet Engineering Task Force (IETF), 2011.
59. A. Musaddiq, Y. B. Zikria and S. W. Kim, "Energy-Aware Adaptive Trickle Timer Algorithm for RPL-based Routing in the Internet of Things," *2018 28th International Telecommunication Networks and Applications Conference (ITNAC)*, Sydney, NSW, 2018, pp. 1–6.
60. B. Djamaa, M. Richardson, "Optimizing the Trickle Algorithm", *IEEE Communications Letters*, vol. 19, no. 5, pp. 819-822, May. 2015.
61. F. Al-Turjman, H. Zahmatkesh and L. Mostarda, "Quantifying Uncertainty in Internet of Medical Things and Big-Data Services Using Intelligence and Deep Learning", *IEEE Access*, vol. 7, no. 1, pp. 115749-115759, Jul. 2019.
62. R. Ali, Y. A. Qadri, Y. B. Zikria, T. Umer, B. Kim, and S. W. Kim, "Q-learning-enabled channel access in next-generation dense wireless networks for IoT-based eHealth systems," *EURASIP Journal on Wireless Communications and Networking*, vol. 2019(178), pp. 1-12, Jul. 2019.
63. F. Al-Turjman, A. Malekoo, "Smart Parking in IoT-enabled Cities: A Survey", *Elsevier Sustainable Cities and Societies*, vol. 49, pp. 101608, Aug. 2019.
64. Z1 Datasheet. (Accessed on 20/11/2019). [Online] Available: [http://zolertia.sourceforge.net/wiki/images/e/e8/Z1\\_RevC\\_Datasheet.pdf](http://zolertia.sourceforge.net/wiki/images/e/e8/Z1_RevC_Datasheet.pdf).
65. Tmote Sky Datasheet. (Accessed on 20/11/2019). [Online] Available: <http://www.crew-project.eu/sites/default/files/tmote-sky-datasheet.pdf>.

## Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Submit your manuscript to a SpringerOpen<sup>®</sup> journal and benefit from:**

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

---

Submit your next manuscript at ► [springeropen.com](http://springeropen.com)

---