Ad Hoc Networks 65 (2017) 78–90

Contents lists available at ScienceDirect

Ad Hoc Networks

journal homepage: www.elsevier.com/locate/adhoc

Design and analysis of RPL objective functions for multi-gateway ad-hoc low-power and lossy networks

Muhammad Omer Farooq[a,*], Cormac J. Sreenan[a], Kenneth N. Brown[a], Thomas Kunz[b]

[a] CTVR, Department of Computer Science, University College Cork, Ireland
[b] Department of Systems and Computer Engineering, Carleton University, Ottawa, Canada

ARTICLE INFO

Article history:
Received 11 February 2017
Revised 6 June 2017
Accepted 4 August 2017
Available online 5 August 2017

Keywords:
RPL
Multi-gateway LLN
Routing protocols
Low-power and lossy networks
IEEE 802.15.4

ABSTRACT

RPL is a standard routing framework for low-power and lossy networks (LLNs). LLNs usually operate in challenged conditions, therefore RPL can be adapted to satisfy requirements of a particular LLN. RPL facilitates this through objective functions (OFs). An OF is used to discover and maintain data forwarding paths based on the requirements of LLNs. In RPL, different OFs can use different routing metrics in different ways. In this paper, we design different OFs and analyse their impact on RPL performance in multi-gateway ad-hoc LLNs. In conjunction with the shortest hop-count, our designed OFs also use the following tie-breaking metrics: available bandwidth, delay, buffer occupancy, and ETX. Our OFs use the tie-breaking metrics on a greedy or an end-to-end basis. In our experimental analysis, we consider the impact of duty-cycling, number of gateways, and data traffic load on the OFs' performance. Our results demonstrate that, generally speaking, the performance improves with an increase in the number of gateways. In the absence of duty-cycling, the greedy approach is better compared to the end-to-end approach, and using delay, buffer occupancy, and ETX metrics as the tie-breaking metrics in conjunction with the shortest hop-count metric yield the best performance. In a relatively high data traffic load, all OFs perform similarly. In duty-cycling mode, frequent changes in the parent node incur extra synchronization time between a sender and receiver. OFs that use the tie-breaking metrics on an end-to-end basis do not frequently change parent nodes, hence they demonstrate better performance. Furthermore, in duty-cycling mode, the shortest hop-count metric demonstrates the best performance compared to the other metrics.

1. Introduction

A low-power and lossy network (LLN) is composed of low-power wireless nodes and one or more gateway nodes. The gateway connects the LLN nodes to the Internet. The nodes are wirelessly interconnected with each other and with the gateways. Such networks are characterized as low-power and lossy networks (LLNs) because nodes possess limited power and they operate in harsh environments. The harsh environments usually cause packet losses and temporary link failures. There are many applications of LLNs, including, e.g., environment monitoring, surveillance, traffic monitoring, industrial process control, home automation and assisted living, using sensors of many different types [1,2]. Nodes capture the data of interest and report it to the gateway. If a node is not in direct communication range of the gateway, the data is reported in a multi-hop manner. Nodes closer to a gateway relay data of those nodes that are further from the gateway, and hence hotspots can occur near the gateway. These hotspot nodes tend to deplete their energy faster, which reduces LLN lifetime. Recent studies demonstrate that using multiple gateways inside a LLN can improve the network's performance and lifetime [3–5].

Depending on the application, data generated by nodes can have different end-to-end packet delivery delay and reliability requirements. For example, a LLN deployed for industrial process control can have stringent delay and reliability requirements, whereas a network deployed for video-surveillance has less stringent delay and reliability requirements. A routing protocol forwards data packets from nodes to any of the gateways, therefore the routing protocol plays a pivotal role in delivering data to the gateway. Considering the characteristics of LLNs and their possible applications, the Internet Engineering Task Force (IETF) ROLL (Routing Over Low-power and Lossy networks) working group standardized the routing architecture for low-power and lossy networks

* Corresponding author.
E-mail addresses: omer.farooq@insight-centre.org, omerfarooq@rocketmail.com (M.O. Farooq), cjs@cs.ucc.ie (C.J. Sreenan), k.brown@cs.ucc.ie (K.N. Brown), tkunz@sce.carleton.ca (T. Kunz).

http://dx.doi.org/10.1016/j.adhoc.2017.08.002
1570-8705/

called RPL. The salient design feature of RPL is a routing framework that allows the use of different routing metrics and objective functions (OFs) to cope with LLNs' limitations and satisfy heterogeneous application requirements. Therefore, an OF is used by RPL to discover and maintain data forwarding paths based on the requirements of LLNs.

We present different RPL objective functions (OFs) for multi-gateway ad-hoc LLNs. Our OFs use the available bandwidth, delay, MAC layer buffer occupancy (the number of frames in the MAC layer queue), and expected transmission count (ETX) as tie-breaking routing metrics in conjunction with the shortest hop-count metric. Our OFs can use the metrics on either a greedy or an end-to-end basis. The following are our main contributions:

1. Design of different RPL OFs for multi-gateway ad-hoc LLNs.
2. Analysing the impact of duty-cycling, number of gateways, and data traffic load on the performance of our designed OFs.
3. Our extensive experimental results demonstrate the following: (*i*) In the absence of duty-cycling, using the different tie-breaking metrics in a greedy manner shows performance improvement compared to using them on an end-to-end basis, (*ii*) for relatively high data traffic loads, our designed OFs perform similarly, and (*iii*) in duty-cycling mode, infrequent parent node switching results in performance improvement, hence here an end-to-end approach is better, and the shortest hop-count metric demonstrates the best performance.

The rest of this paper is organised as follows: a description of RPL is presented in Section 2, related work is presented in Section 3, our RPL OFs and corresponding routing protocols for multi-gateway LLNs are presented in Section 4, performance evaluation is presented in Section 5, and dummyTXdummy- finally our conclusions are given in Section 6.

2. RPL: routing in low-power and lossy networks

RPL is a proactive distance vector routing protocol for LLNs [6]. The protocol operates at the networking layer, hence it can support multiple link layer technologies. RPL supports multi-point to point (nodes to the gateway[1]), point to multi-point (gateway to nodes), and peer-to-peer (node to node) communication. For route construction RPL uses the concept of destination oriented directed acyclic graph (DODAG), and it uses the following control messages:

(1) DIO: DODAG Information Object
(2) DIS: DODAG Information Solicitation
(3) DAO: Destination Advertisement Object

The main purpose of the DIO message is to build a DODAG rooted at the gateway. The DIS message is used to solicit a DIO from a RPL node, it is normally send by a node when it joins a stable network. The DAO message is used to construct routes from gateways to nodes and from nodes to nodes, it contains prefix reachability information.

The RPL standard writes the following: "Most implementations are required to support no downwards routes, non-storing mode only, or storing mode only" [6]. Therefore, this paper focuses on upward communication.[2] Our work can be easily extended to support use cases that require downward and peer-to-peer communications. However, remainder in this section focuses on upward route construction and maintenance in RPL.

[1] As per the RPL nomenclature the gateway is referred to as the root, but for the sake of consistency we use the word gateway instead of the root.

[2] communication from nodes to the gateway.

2.1. DODAG construction

Initially, a gateway periodically multicasts a DIO message. Nodes in the transmission range of the gateway receive the DIO message and decide to join the DODAG based on their OF. If the nodes join the DODAG, the nodes periodically re-multicast the message. The process repeats at each node, and allows nodes to select their parent nodes towards the gateway. Note that leaf nodes only join the DODAG, but do not multicast the message. There can be multiple DODAGs inside a network, and they are differentiated by their instance ID. The idea is that, if a node's OF is to forward data packets on a data forwarding path that offers highest reliability, it joins the DODAG that offers highest reliability. Similarly, there can be another node whose OF is to forward critical data on a path that offers highest reliability, and at the same time forward real-time multimedia data on a path that offers least delay. In this case, the node joins two DODAGs: one that offers highest reliability and another one that offers least delay. A single DODAG is called a RPL instance. A node can join multiple DODAGs with different IDs, but it can only join a single DODAG with the same ID. A node can switch between DODAGs with the same ID, but in that case the node has to abandon its current parent.

2.2. Routing metrics and constraints support

Because of the diverse applications of LLNs and their energy, processing, size, and memory limitations, it is impractical to select a single or a combination of routing metrics for all applications. Therefore, the RPL specification does not fix any metric, rather it is left to the discretion of a network designer/network administrator to choose a metric that best suits the purpose. Moreover, RPL allows pruning of nodes and links from a path using constraints, e.g., it avoids links with a signal-to-noise ratio below a certain threshold.

2.3. Loop avoidance and detection

RPL does not guarantee loop-free routing, but it tries to avoid and detect them. In RPL each node has a rank, and it is a node's relative position from the gateway. To avoid loops, the RPL standard specifies two rules: max-depth and greedy. In the max-depth rule, a node is not allowed to select a deeper parent node, such that the node's rank becomes greater than max-depth. Max-depth is a configurable parameter at the gateway. In the greedy rule, a node cannot move deeper in the graph to increase the number of parents. Loop detection is achieved by setting bits in the RPL routing header. For example, if a node sends a data packet to its child, the node sets the down bit in the header. Upon receiving the packet with the down bit set, the child can infer a loop if, after performing the routing table lookup, it learns that the packet needs to be forwarded upward.

2.4. Route repair

In case of node or link failures, RPL can use the following two methods for route repairs: local repair and global repair. In the local repair, if a node detects link or node failure, the node tries to repair the route by routing through a sibling with the same rank or the node switches parent. The global repair can only be initiated by the gateway, therefore it incurs additional control message overhead. The gateway can initiate the global repair if it receives an inconsistent identifier for the DIO message.

2.5. Frequency of DIO messages

LLN contains nodes with limited resources, therefore it is essential to limit the amount of control packets. RPL broadcasts DIO

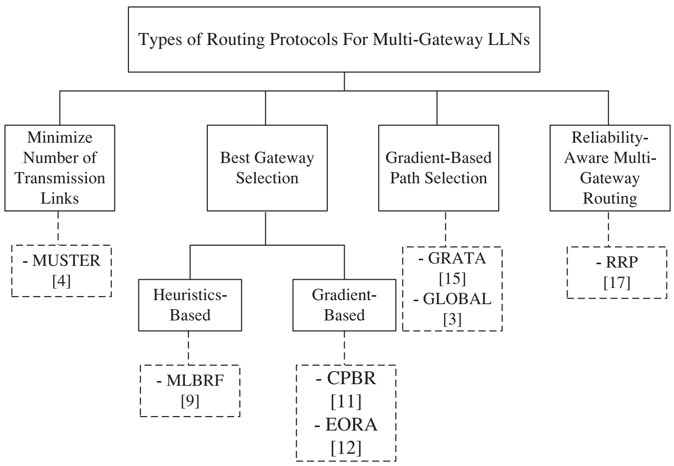

Fig. 1. Routing protocols for multi-gateway LLNs.

messages using a trickle timer. DIO messages are broadcasted more frequently in any of the following situations: the network is not stable, inconsistency in the network, and a new node joins the network. As the network becomes more stable the DIO broadcast frequency reduces till it reaches a predefined value.

3. Related work

Fig. 1 shows different categories of routing protocols for multi-gateway WSNs along with some existing routing protocols in each category.

Minimize transmission links: Routing protocols presented in [4,7] try to minimize the number of transmission links by maximizing the overlap among different data forwarding paths to multiple gateways. A node executes a quality function corresponding to its one-hop candidate downstream neighbours. The inputs to the function are: distance of the neighbour node to each gateway, number of different source-gateway flows passing through the neighbour, and number of gateways that can be served by the neighbour. Based on the values of the function, the minimum number of neighbours required to serve all the gateways are selected as parent nodes. The function is re-evaluated after a pre-defined time interval. The input to the function corresponding to the neighbours is gathered using the following methods: piggybacked on application messages and overhearing during transmissions. The drawback of the protocols is that, in maximizing the overlapping among different paths, congestion can occur. Mostly, congestion results in a higher end-to-end delay and lower packet delivery ratio (PDR).

Heuristics-based best gateway selection: Routing protocols presented in [8,9] use fuzzy algorithms to select the best gateway for data packets at a source node. The protocols are designed to satisfy any one or both of the following objectives: minimize energy consumption and maximize reliability. Depending upon an application's requirements, the input to the algorithms is a proper subset of the following: number of one hop candidate downstream nodes leading to a gateway, number of one hop neighbours of the downstream node leading to the gateway, remaining energy of the downstream nodes, distance of the downstream nodes to the gateway, and buffer occupancy at downstream nodes. Nodes periodically advertise the information required by the algorithms. The algorithms are periodically executed at nodes considering all gateways. The best gateway is selected based on the output of the algorithms. The protocols' drawback is the localized decisions making, i.e., the state of the data forwarding paths is not considered on an end-to-end basis.

Gradient-based best gateway selection: The routing protocols presented in [10–14] construct a gradient field based on any one or a combination of the following metrics: hop-count, one hop downstream neighbours' energy level, neighbours' buffer occupancy level, neighbours' node-traversal delay, end-to-end energy level on a data forwarding path, and end-to-end delay on the path. Based on the metric, gradient fields to all gateways are constructed. The information required to construct the gradient fields are either broadcasted periodically or when there is a substantial change in the value of the metric. If a source node has a data packet to transmit, it selects the gateway to which it has the steepest gradient. Relaying nodes forward the packet on a path that offers the steepest gradient to the gateway. The drawbacks of the protocols are: if protocols construct the gradient field locally, the gradient field may not be optimal on an end-to-end basis, and the protocols that construct the gradient field by only using end-to-end energy-level or delay metric may end up selecting longer paths. Longer paths result in a higher delay and lower PDR.
Gradient-based best path selection: The routing protocols presented in [3,5,15,16] construct and maintain a best data forwarding path towards all gateways. This is done assuming an application selects the gateway, hence a routing protocol does not select the gateway node for a data packet (in general, this is the only difference compared to gradient-based best gateway selection category). Gradient fields towards gateways are constructed using a combination of the following metrics: shortest hop-count, residual energy of one hop downstream nodes, downstream node's mean buffer occupancy, maximum buffer occupancy at two hop downstream nodes, and end-to-end energy depletion rate. A node periodically broadcasts the information required to construct the gradient field. The protocols' drawbacks are similar to the drawbacks of the protocols discussed in the gradient-based best gateway selection category.

Reliability-aware multi-gateway routing: The routing protocols presented in [17,18] aim to increase reliability. In [17], the routing protocol attempts to discover and maintain two disjoint data forwarding paths to each gateway. Forwarding the same data packet on the two paths increases reliability as error probabilities on the paths are independent. In [18], the routing protocol constructs an energy-efficient minimum spanning tree towards K gateways among a total of M gateways in a network, and $K < M$. To increase reliability a data packet is forwarded to K gateway nodes. Forwarding the same data packets on multiple paths incurs extra energy and can cause congestion in a network.

RPL-based routing protocols:*a* The routing protocols presented in [19,20] adapt RPL to support mobility. The protocol presented in [19] assumes a hybrid network, i.e., some nodes in a network are mobile and some are stationary. As node mobility can lead to frequent route failures, therefore the protocol partially addresses this problem by restricting mobile nodes to act as leaf nodes. The leaf nodes in RPL cannot act as relays, therefore their mobility does not result in route failures for other nodes. However, due to the protocol design and mobility, isolated networks can emerge in the network. In [20], a corona mechanism is used to adapt RPL to support mobility. The network area is partitioned into multiple concentric circular regions centred at the root. Each circular region has its own unique corona ID, the circular region closer to the root has lower circular ID compared to the circular region away from the root. With mobility nodes update their corona ID. The nodes select their parent nodes with lower corona ID compared to their own corona ID. In [21], the RPL protocol is adapted to support cluster-based WSNs with mobile gateway nodes. Using the protocol, sensor nodes discover forwarding paths to cluster heads, and the cluster heads discover paths to the mobile gateways. Hence, the sensor nodes only communicate with cluster heads, and the cluster heads communicate with the gateways. The protocol defines new control messages to provide mobility support with RPL. As the gateways are mobile, the protocol also defines mechanism to indicate lifetime of a forwarding path to the gateways. A link quality indicator is used as the routing metric. The routing protocol presented in [22] is an adapted version of RPL, while discovering data forwarding paths, the protocol considers the resource heterogeneity of nodes in a network. For this purpose, the protocol presents a routing metric that considers available resources at nodes in a network. As the state of resources changes so does the value of the metric. Nodes select parent nodes based on the best value of the metric, hence the routing protocol selects data forwarding paths with better resources. The protocol also helps to drop low-resources nodes from the data forwarding paths.

In [23] an energy-efficient region-based RPL routing protocol is presented. The protocol partitions a sensing field into different regions. Each region has a dedicated reference node (a node equipped with a global positional system), and each node in a network discovers the reference node in its region. For multipoint-to-point routing the algorithm uses the RPL proactive routing mechanism. For peer-to-peer (P2P) routing, the protocol uses a reactive approach, and nodes participate in the route discovery procedure based on the knowledge of source and destination nodes' regions. This mechanism helps to reduce the control message overhead as only a subset of nodes participate in the discovery of P2P routes. The protocol cannot work without local information. In [24], a distributed algorithm is presented to quickly detect the DODAG root failure, and the algorithm improves RPL performance.

In [25], RPL performance has been evaluated using different OFs. But the performance evaluation does not consider the impact of radio duty cycling algorithm on the different OFs' performance. The control message overhead associated with the OFs is not shown. Moreover, the analysis of the presented results lacks in some aspects, for example, the impact of extra control messages multicast on a node's actual data packets transmission capability is not analysed.

The performance of RPL has been evaluated for a single gateway network in [26–28] mostly using the hop-count and/or ETX metrics, and RPL is used for multi-gateway network in [29]. As RPL is a routing framework for LLNs, therefore based on the framework, we can design different routing protocols by changing routing metrics and OFs. An interesting research question in this context is to design different OFs and analyse their impact on a network's performance in terms of the following: PDR, packet delivery delay, forwarding path length, and energy consumption. As RPL is designed for LLNs, therefore another research question is to analyse the impact of radio duty-cycling on a performance of different RPL OFs. A single gateway in a relatively large network can become the performance bottleneck, therefore yet another research question is to analyse different OFs performance in a multi-gateway network. The answers to the stated research questions can help to determine an OF or a set of OFs which can yield better RPL performance in multi-gateway LLNs. Therefore, the purpose of this paper is to obtain answers to the stated research questions.

4. Objective functions and routing protocols for multi-Gateway ad-hoc LLNs

Industrial IoT, remote health monitoring, assisted living, sharing multimedia content in home/office environment, and smart home are some of the many LLN use cases. Some of the use cases require high reliability and other require bounded delay and/or bandwidth guarantee. In case of a single gateway in such a network, data traffic hot spots can occur near the gateway. Thus, the single gateway can possibly become a cause of congestion, and the congestion can result in lowering reliability and increasing latency. Furthermore, in a single gateway-based LLN, the mean forwarding path length from any node in a network to the gateway will increase. This can result in reduced reliability and increased latency. Multiple gateways can possibly help to reduce hot spots and mean forwarding path length, thus can help to increase reliability and reduce latency.

In this section, we describe the following regarding the design of RPL-based routing protocols for multi-gateway ad-hoc LLNs: routing metrics, OFs, DODAG construction and data packet forwarding, and protocol overheads.

4.1. Routing metrics

Our RPL-based routing protocols use routing metrics related to throughput, delay, and reliability. We use available bandwidth as a representative of throughput-based metrics, delay and MAC layer queue occupancy as representatives of delay-based metrics, and ETX as a representative of reliability-based metrics. The metrics are used as the tie-breaking metrics in conjunction with the shortest hop-count metric. In the rest of this sub-section we discuss the methods used to calculate values of these metrics.

Available bandwidth. Available bandwidth is an indication of a communication link's residual data relaying capacity. High available bandwidth implies low data load on the link, hence the link may contribute in achieving low delay and high PDR. To estimate the available bandwidth we use the algorithm presented in [30]. For the readers convenience, we briefly summarize this algorithm here. Using control messages, a node keeps track of the data generation rates of nodes within the node's interference range. The IEEE802.15.4's CSMA-CA MAC protocol also consumes bandwidth, e.g., a node cannot transmit while it is in back-off mode or waiting for an ACK. Therefore, the algorithm keeps track of the bandwidth consumed by the MAC layer operation per unit time. The MAC layer overhead measure in time is converted to bps by multiplying the overhead with the channel rate. To cope with any wireless channel impairments (reflection, refraction, and multi-path fading) the algorithm uses sliding-window-based averaging to estimate the available bandwidth, and Eq. (1) is used for available bandwidth estimation.

$$\omega_n = \rho - \left(\frac{\sum_{\mu=1}^{\theta} \beta_\mu + \gamma_\mu}{\theta} \right) bps \quad (1)$$

In Eq. (1), ω_n denotes the average available bandwidth in bps at any node n, θ denotes the current size of the averaging window (the maximum value of θ is α, and through experiments it is shown in [31] that 5 is a suitable value for α), β_μ denotes the total data generation rate within the interference range of the node at the μth index of the averaging window, γ_μ denotes the total MAC layer overhead at the μ^{th} index of the averaging window, and ρ denotes the channel rate.

Delay. The time spent by a data packet in the MAC layer queue impacts the end-to-end packet delivery delay and PDR, therefore delay is an important routing metric. To obtain the delay, the following method is used. The time when a data packet was enqueued in the MAC layer queue is subtracted from the time when the packet was successfully transmitted to obtain the delay incurred in transmitting the packet. The delay of each packet is accumulated per unit time to obtain total delay. Finally, the delay is obtained by dividing the total delay with the total number of packets transmitted per unit time. We use a time unit of 1 s. The algorithm uses the sliding-window-based averaging with a window size of 5 s to obtain the node traversal delay.

MAC layer queue occupancy. Transmitters and receivers are not synchronized in an ad-hoc wireless network. Therefore, there can be time instances when the delay at nodes with lower data generation rates can be relatively higher. The delay metric may select a parent node which is already generating data at a higher rate. This can lead to congestion, to avoid such scenarios the MAC layer queue occupancy metric can be used. If a routing protocol successfully avoids congested nodes, it can demonstrate good results. The number of frames in the MAC layer queue are sampled per unit time. The sliding-window-based averaging with a window size of 5 s is used to obtain the MAC layer queue occupancy.

Expected transmission count (ETX). ETX is the expected number of transmissions required by a data packet to be delivered successfully. ETX is the ratio of the total transmission attempts (including retransmissions) to the total number of packets delivered successfully per unit time. An ETX value of one indicates a perfect communication link, and the higher the ETX value the lower the quality of the communication link. Therefore, using the ETX metric can help to select a data forwarding path that includes relatively high quality communication links. High quality communication links imply fewer retransmissions, hence higher PDR and lower delay and energy consumption. In our implementation, ETX at a node is calculated every second (if the node is transmitting data packets), and we use the sliding-window-based averaging with a window size of 5 s to obtain mean ETX at a node.

4.2. Objective functions

4.2.1. Notation

If x is a tuple $(x_0, x_1, x_3, ..., x_n)$ then the length of x, $|x| = n + 1$. We can write x as $(x_0|x')$ where $x' = (x_1, x_2, ..., x_n)$. Let λ be an empty tuple, and so $|\lambda| = 0$ and $\lambda = ()$. We define $\leq_{lex}$ as follows:

$$\begin{cases} x \leq_{lex} y \iff x = \lambda & \vee \\ x = (x_0|x'), y = (y_0|y') \wedge & x_0 < y_0 \vee \\ (x_0 = y_0 \wedge x' \leq_{lex} y') & \end{cases} \tag{2}$$

Let n be a node, $N(n)$ is the set of neighbours of n, $S(n)$ is the hop-count from n to the gateway, $a(n)$ is the available bandwidth metric for n, $d(n)$ is the delay metric for n, and $b(n)$ is the MAC layer buffer occupancy metric for n. Similarly, $a'(n)$ is the minimum end-to-end available bandwidth to a gateway at n, $d'(n)$ is the end-to-end delay to a gateway at n, and $b'(n)$ is the maximum end-to-end buffer occupancy to a gateway at n.

Our RPL-based routing protocols are based on one of the following OFs:

1. Objective function 1 (OF1). Discover and maintain data forwarding paths to gateways using the shortest hop-count routing metric. In case there are multiple such paths, randomly select one.

   $$\textit{Minimize}\, S(z)\ \textit{where}\, z \in N(n) \tag{3}$$

   The above expression means choose the neighbour z with minimal hop-count to the gateway.
2. Objective function 2 (OF2). Discover and maintain data forwarding paths to gateways using the shortest hop-count routing metric. In case there are multiple such paths, select the one on which a candidate parent node has advertised better value of the tie-breaking metrics (available bandwidth, delay, MAC layer buffer occupancy, and ETX). If there are more than one such candidate parents, randomly select one parent. OF2 is based on a greedy approach.

   $$\textit{Minimize}_{\leq_{lex}} (S(z), a(z))\ \textit{for}\, z \in N(n) \tag{4}$$

   The above expression means choose the neighbour z with minimal hop-count, and if there are many, choose the one with minimum available bandwidth. Similarly, for other tie-breaking metrics we can write the following:

   $$\begin{cases} \textit{Minimize}_{\leq_{lex}} (S(z), d(z))\, \forall z \in N(n) \\ \textit{Minimize}_{\leq_{lex}} (S(z), b(z))\, \forall z \in N(n) \end{cases} \tag{5}$$
3. Objective function 3 (OF3). Discover and maintain data forwarding paths to gateways using the shortest hop-count routing metric. In case there are multiple such paths, select the one on which a candidate parent node has advertised a better end-to-end (from candidate parent to gateway) value of the tie-breaking metrics (available bandwidth, delay, MAC layer buffer occupancy, and ETX). If there are more than one such candidate parents, randomly select one parent. OF3 is based on an end-to-end approach.

   $$\textit{Minimize}_{\leq_{lex}} (S(z), a'(z))\, \forall z \in N(n) \tag{6}$$

   The above expression means choose the neighbour z with minimal hop-count, and if there are many, choose the one that has advertised better end-to-end available bandwidth. Similarly, for other tie-breaking metrics we can write the following:

   $$\begin{cases} \textit{Minimize}_{\leq_{lex}} (S(z), d'(z))\, \forall z \in N(n) \\ \textit{Minimize}_{\leq_{lex}} (S(z), b'(z))\, \forall z \in N(n) \end{cases} \tag{7}$$

4.3. DODAG construction and data forwarding

For DODAG construction DIO messages are used in the same way as described in Section 2.1. For a detailed explanation about the message structure readers are encouraged to read [6]. Different DODAGs are identified using the RPL instance ID and DODAGID (gateway node network layer address). The *rank* field of the message contains the hop-count to the gateway. To advertise the value of any one of the tie-breaking metrics, we use the options field of the message, and six additional bytes are used to store type, length, and the metric value in the message. The gateways are represented by set S. An element in the set S is denoted by s_i. Each node maintains a routing table, and a record in the routing table stores the following information about the discovered gateways: gateway id ($s_{\text{i-id}}$), RPL instance ($rpl_{instance}$), parent ($s_{\text{i-parent}}$), rank ($s_{\text{i-rank}}$), tie-breaking metric value ($s_{\text{i-tie}}$), and a *joined* flag that shows whether the node has joined the DODAG or not. In the following discussion an instance of the DIO message is denoted by *dio*. Moreover, RPL instance, rank, a tie-breaking metric value, and gateway address in the message are denoted by $\text{dio.rpl}_{\text{instance}}$, $\text{dio.s}_{\text{i-rank}}$, $\text{dio.s}_{\text{i-tie}}$, and $\text{dio.s}_{\text{i-id}}$ respectively. If a RPL instance uses the value of a tie-breaking metric on the greedy basis, the instance corresponding to the available bandwidth, delay, MAC layer queue occupancy, or ETX is identified by the values 1, 2, 3, and 4 respectively. Moreover, if the instance uses the value of a tie-breaking metrics on an end-to-end basis, the instance corresponding to the available bandwidth, delay, MAC layer queue occupancy, or ETX is identified by the values 5, 6, 7, and 8 respectively. If RPL uses shortest hop-count metric, the instance is identified by the value 9. If $\text{dio.rpl}_{\text{instance}}$ is 9, $\text{dio.s}_{\text{i-tie}}$ is always set to 0.

Initially, for all gateways, $s_{\text{i-rank}}$ and $s_{\text{i-tie}}$ are set to ∞. Furthermore, $s_{\text{i-tie}}$ corresponding to the available bandwidth is set to 0. The list of a node $n's$ OFs is represented by set $inst_n$. An item in the set $inst_n$ is denoted by $inst_{n_i}$. For OF1 $inst_{n_i}$ can only take the value 9 (the value of the RPL instance for the hop-count metric). For OF2 and OF3 $inst_{n_i}$ can take any value in the range [1, 4] and [5, 8] respectively. The size of set $inst_n$ is denoted by $size_n$. The node that broadcasted the DIO message is denoted by *dio.src_addr*.

Algorithm 1 summaries the DODAGs construction and maintenance. When a node receives the DIO message, the node checks whether it is interested in joining the DODAG. If so, the node joins the DODAG in the following cases: the message contains a DODAG to a new gateway or the advertised DODAG is better than the existing DODAG. Afterwards, the node updates its routing table, if required. The source node selects the nearest gateway node in terms of the hop-count to transmit data packets.

Algorithm 1: DODAGs construction and maintenance.

```
Input: dio;
routingRecord rt − rec;
i ← 0;
node − interested ← req − to − join ← false;
while i ≤ size_n do
    if inst_{n_i} == dio.rpl_instance then
        node − interested ← true;
        break;
    end
    else
        i = i + 1;
    end
end
rt − rec = search_rt_table(dio.rpl_instance, dio.s_{i−id});
if rt − rec == NULL then
    insert_rec_in_rt_table(dio);
    if node − interested then
        join_DODAG(dio);
    end
end
else
    if rt − rec.s_{i−rank} > dio.s_{i−rank} then
        rt − rec.s_{i−rank} ← dio.s_{i−rank};
        rt − rec.s_{i−parent} ← dio.src_addr;
        rt − rec.s_{i−tie} ← dio.s_{i−tie};
        req − to − join ← true;
    end
    else
        if (dio.rpl_instance > 0) && (dio.rpl_instance < 9) then
            if rt − rec.s_{i−rank} == dio.s_{i−rank} then
                if is_better(dio.s_{i−tie}, rt − rec.s_{i−tie}) then
                    rt − rec.s_{i−parent} ← dio.src_addr;
                    rt − rec.s_{i−tie} ← dio.s_{i−tie};
                    req − to − join ← true;
                end
            end
        end
    end
    if (node − interested) && (req − to − join) then
        join_DODAG(dio);
    end
end
```

Periodically, a node broadcasts each DODAG it has joined in the DIO message. In the rank field of the message, the node advertises its hop-count to the gateway. In the option field, the node advertises the value of the tie-breaking metric being used. If the DODAG is based on OF2, the node advertises its locally calculated value of the metric. Otherwise, the value that reflects the end-to-end DODAG status is inserted. For example, if the available bandwidth metric is used, the minimum of the node's own available bandwidth and the available bandwidth advertised by the node's parent determines the node's advertised bandwidth.

4.4. Protocol control overheads

There are two kinds of overheads for DODAG construction: the DIO message overhead and the overhead for calculating the value of the routing metrics. As delay, the MAC layer queue occupancy, and ETX can be determined by a node locally, there is no overhead associated with them. But, for estimating the available bandwidth, a node is required to know the available bandwidth and transmission rates of nodes per unit time within its interference range, therefore a control message is required to estimate the available bandwidth [30]. Eq. (8) can be used to determine network-wide mean control bits overhead per unit time. In Eq. (8), $\overline{T}$ is the mean number of nodes within the interference range of a node, j is the total number of neighbour information structures that can be carried in a single message, n is the number of nodes inside a network, l is the size of the neighbour information structure, and i is the size of the message header. A neighbour information structure holds neighbour's information, i.e., neighbour id, transmission rate, and available bandwidth.

$$OH = \begin{cases} \left(n \times (\overline{T} \times l)\right) + (n \times i) & \overline{T} \le j \\ \left(n \times (\overline{T} \times l)\right) + \left(\lceil \frac{\overline{T}}{j} \rceil \times (n \times i)\right) & \overline{T} > j \end{cases} \quad (8)$$

The frequency of DIO messages depends on the rate at which the value of a routing metric changes, i.e., if the value changes fast, the message should be send more frequently. On the other hand, the messages should be send at a pre-defined minimum rate. Moreover, a threshold (*TH*) for the available bandwidth, delay, MAC layer buffer occupancy, and ETX can be defined, and once a network is in a stable state, the message is only transmitted if there is *TH* change in the value of the metric or the maximum time between the two successive messages transmission has elapsed. Deriving an appropriate value for *TH* is beyond the scope of this paper. In our experiments DIO messages are transmitted every second.

5. Performance evaluation

In this section, we evaluate different OFs' performance. For a detailed performance evaluation, we study the impact of the number of gateway nodes in a network, data traffic load, and radio duty-cycling on the protocols' performance. In this section, we categorize our performance benchmarks as follows: reliability, latency, and energy consumption. For reliability we measure and report mean PDR, for latency we report mean per-packet end-to-end delay, and for energy consumption we measure total retransmissions averaged across total simulation runs and protocols' control overhead.

Simulations were performed using the widely used Cooja WSN simulator [32] that uses real programming code for a wireless sensor node. We used a grid network topology with 75 nodes placed in a 300×300 m^2 area. Based on published work we vary the number of gateways from 2 to 4 [3,5,11,15], and gateways are randomly placed in the network. Each node generates data packets, and the packet generation rate is randomly distributed in the range [1, 3] packets/second, and the size of data frame is 127 bytes. Nodes generate packets using an on/off schedule, i.e., the nodes generate the packets for a duration randomly distributed in the range [2, 5] seconds, afterwards the nodes wait for a random duration of time distributed in the range [10, 15] seconds before generating packets again. No node generates packets after 100 simulation seconds. The total duration of a single simulation is 115 s. Our traffic generation model is a representation of a data traffic generated by a range of event-detection system, i.e., upon detecting an event a small burst of packets is transmitted. For example, a LLN can be deployed to monitor traffic on a network of roads. Upon detecting a traffic rule violation a few images of a vehicle is transmitted. Similar traffic models are used by other event detection systems e.g., fire detection, target tracking, etc. Our results are based on 10 simulation runs (randomly placing gateways each time) for each number of gateway nodes. In the following figures, we plot the mean value for each protocol, and we show as error bars the 95% confidence intervals (CIs) around the mean, based on a t-distribution with a sample size of 10. Note that where CIs over-

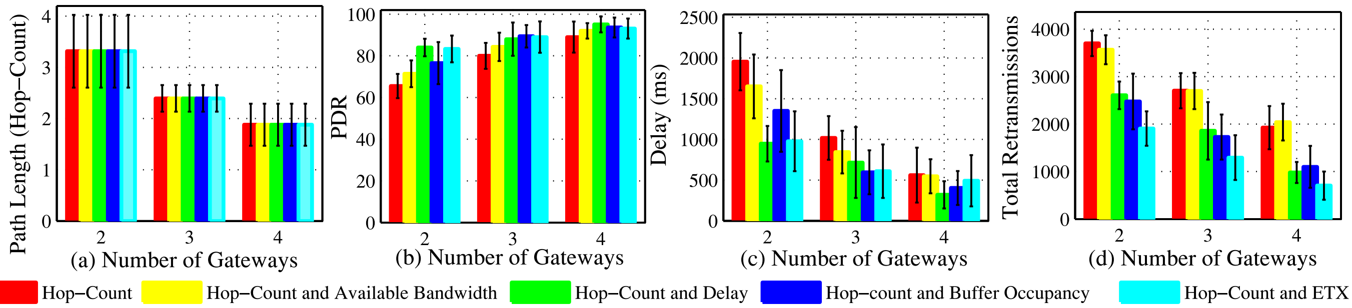

Fig. 2. RPL-based protocol performance using a greedy approach (null RDC).

Table 1
General simulation parameters.

Parameter	Value
MAC layer	IEEE 802.15.4 CSMA-CA
MAC layer reliability	Enabled
Radio model	Unit disk graph with distance loss
Channel rate	250 kbps
MAC layer queue size	20 frames
Node transmission range	50 m
Node carrier sensing range	100 m
Total frame size	127 bytes
Motes emulated	TelosB

lap and means are not in the overlap region, we base our conclusions on the result of a *t*-test. Table 1 shows general simulation parameters.

5.1. Results in the absence of radio duty-cycling

The Protocol performance evaluation without duty-cycling gives an upper bound on the performance. Therefore, to obtain the upper bound on the protocols' performance, we carried out a set of experiments, and the results presented in this section are based on Contiki2.5's Null radio duty cycling (RDC) algorithm.

Fig. 2 shows the performance of different RPL-based protocols using the greedy approach. Fig. 2(a) shows that the mean path length decreases as the number of gateways increases, and the difference is statistically significant. The mean path length for all protocols is the same because candidate parents are selected based on the shortest hop-count, and the tie-breaking metric are used to select the parent in case there is a tie. In general, the mean PDR increases and the mean end-to-end per-packet delay decreases as the number of gateways increases, as shown in Fig. 2(b) and (c) respectively. Mostly, the protocols using delay, queue occupancy, or ETX in conjunction with the shortest hop-count demonstrate better PDR and delay, but the difference is not statistically significant compared to the others. This is due to the following reasons: the protocols select the same length paths and due to the shared nature of the wireless channel, different parents contend for the same channel. Fig. 2(d) compares the mean total retransmissions. Mostly, sensor nodes have limited energy supply, therefore it is important to evaluate the protocols w.r.t. the retransmissions as a higher number of retransmissions implies more energy consumption. It is evident from Fig. 2(d) that the protocols using hop-count and available bandwidth and only hop-count demonstrate a similar number of retransmissions, and the protocols using delay, buffer occupancy, or ETX in conjunction with the hop-count demonstrate a similar number of retransmissions. In addition, the latter set of protocols demonstrate statistically significantly fewer retransmissions compared to the former set of protocols. In case of four gateways the latter set of protocols approximately demonstrate at least 50% fewer retransmissions, and ETX demonstrates 65% fewer retransmissions. In a stable network, nodes do not change their parents using hop-count, therefore contention does not vary much on transmitters along the path. By nature, the available bandwidth metric operates on a channel level, and results in fewer changes in parents, therefore the contention level does not vary much. However, delay, buffer occupancy, and ETX operate on a per-node level, and their values change frequently. This results in frequent changes in parents, hence varied contention on nodes along different paths, which positively impacts the performance of the protocols in terms of total retransmissions.

Fig. 3 compares the routing protocols using a tie-breaking metric on an end-to-end basis. The results shown in Fig. 3 show similar patterns as those discussed in Fig. 2. But, in the case of 2 gateways, the protocols based on delay and buffer occupancy metrics demonstrate approximately 25% higher PDR compared to the hop-count metric. In some cases, the mean values corresponding to PDR, delay, and retransmissions have deteriorated somewhat compared to the same values in Fig. 2. The reason for this is, if the value for the tie-breaking metric deteriorates multiple hops away from the source, a certain amount of time is required to propagate the change in the value to the source, therefore it is possible that for some time a sub-optimal path is being used. Fig. 3 does not plot mean path lengths as they are the same as shown in Fig. 2 (a), and the same is true for the rest of the paper. From the results we can conclude that it is better to use the greedy approach as it demonstrates a slight performance improvement over the end-to-end approach and it does not require monitoring and propagating the tie-breaking metric on an end-to-end basis. Moreover, the protocols using delay, buffer occupancy, or ETX, in conjunction with the hop-count are better, as the protocols demonstrate statistically significantly fewer retransmissions, and delay and buffer occupancy based protocols also demonstrate higher PDR in some cases.

Fig. 4 shows the total number of control messages transmitted by the protocols averaged across all experiments. Hop-count, delay, buffer occupancy, and ETX based protocols demonstrate a similar number of control packet transmissions. However, the total number of control packet transmissions is 95% higher for the available-bandwidth-based protocol compared to the other protocols. The higher number of control packet transmissions is due to the additional control packets required to estimate the available bandwidth.

Fig. 5 shows the total number of data packets transmitted by the protocols. The protocols demonstrate a similar number of data packet transmissions apart from the available-bandwidth-based protocol. The available-bandwidth-based protocol demonstrates approximately 70% fewer transmissions. It has been discussed in [33] that there is a limit on the number of packets that can be transmitted using the Contiki 2.5 Operating System. Hence, the higher control overhead in case of the available-bandwidth-

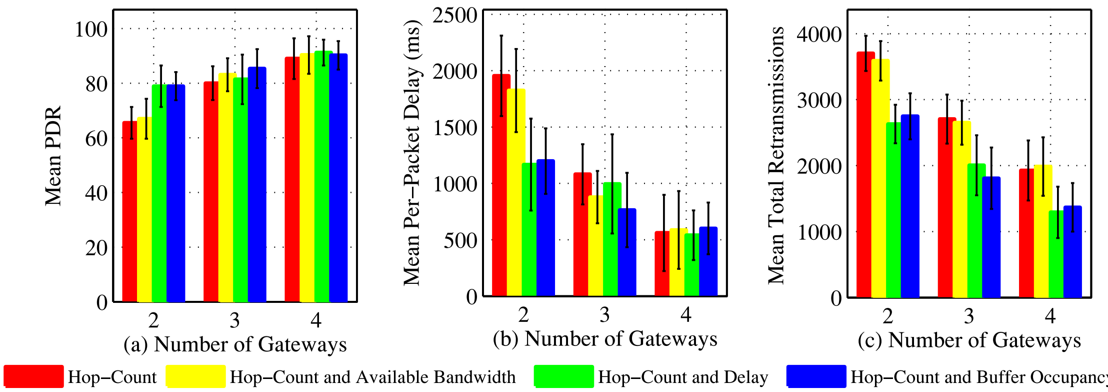

Fig. 3. RPL-based protocol performance using an end-to-end approach (Null RDC).

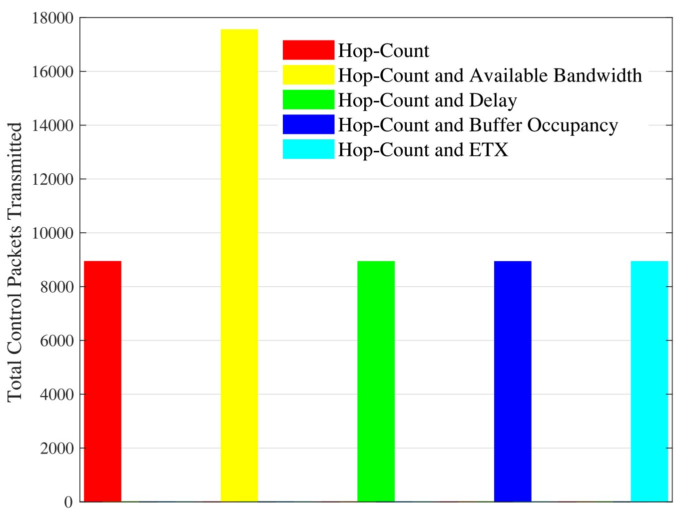

Fig. 4. Control overhead with null RDC.

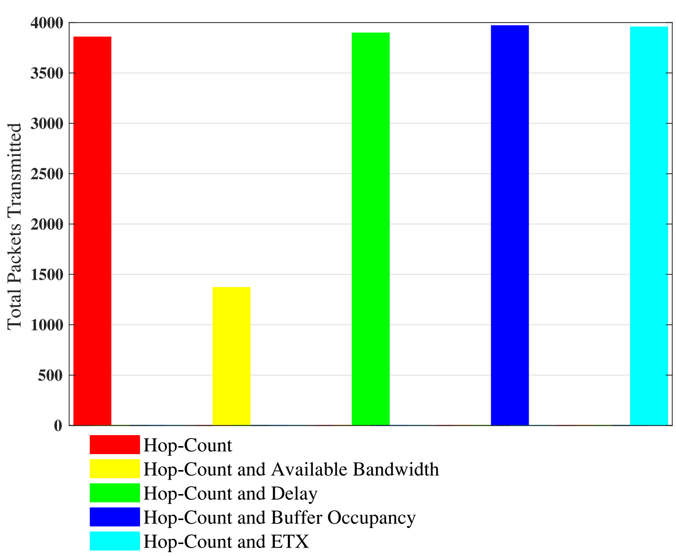

Fig. 5. Total packets transmitted using null RDC.

based protocol negatively impacts the number of data packets transmissions.

In the described set of simulations, we do not change the packet generation distribution as the number of gateways increases. Therefore, we performed another set of simulations by changing the packet generation distribution. We increase the packet generation rate w.r.t. the number of gateways. For 3 and 4 gateways, the packet generation distributions change to [2, 5] packets/sec and [2, 6] packets/second respectively.

Fig. 6 shows the routing protocols' performance using the greedy approach and increased data generation rates. The protocols demonstrate similar performance w.r.t. the recorded metrics. Comparing Fig. 6 with Fig. 2 reveals that the protocols' performance deteriorated with an increase in the data generation rate. All the protocols demonstrated a similar number of retransmissions, however this was not the case in Fig. 2. The increased data transmission inside the network caused congestion, hence higher and similar retransmissions.

Fig. 7 shows the routing protocols' performance using an end-to-end approach and increased data generation rates. The protocols again demonstrate similar performance. Comparing the results presented in Figs. 6 and 7 reveals a random pattern, i.e., in some cases the greedy approach demonstrates a slightly better performance and in other cases the end-to-end approach demonstrates a slightly better performance. The protocols' performance deteriorates compared to the results shown in Fig. 3, hence we can conclude that in a state of network congestion all the protocols perform similarly.

5.2. Results using radio duty-cycling

We carried out another set of simulation-based experiments to analyse the impact of duty cycling on the routing protocols' performance using Contiki2.5's ContikiMAC RDC algorithm [34]. The general simulation parameters and traffic model are the same as described earlier. Fig. 8 shows the protocols' performance using a greedy approach. Fig. 8 (a) and (b) demonstrate that, in general, the mean PDR increases and the mean end-to-end per-packet delay decreases as the number of gateway increases. This is consistent with the results shown in Fig. 2 (b) and (c). Among the evaluated protocols the hop-count-based protocol demonstrates higher PDR and lower delay and total retransmissions. The remaining protocols demonstrate similar PDR, delay, and total retransmissions. Under the duty cycling operation, the hop-count-based protocol demonstrates the best performance, and this differs noticeably from the case without the duty cycling operation. The other protocols demonstrate poor performance compared to the hop-count-based protocol because of the following reasons:

(1) The other protocols also depend on the tie-breaking metrics' value, and in the duty cycling operation the probability of successful transmission is lower. Therefore, it is likely that nodes receive delayed updates about the tie-breaking metric. Such de-

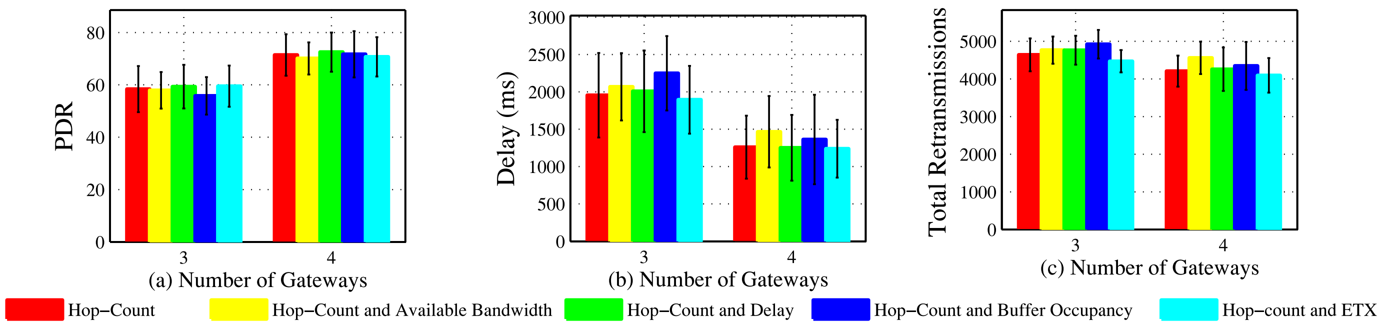

Fig. 6. RPL-based protocol performance using a greedy approach and increased data generation (Null RDC).

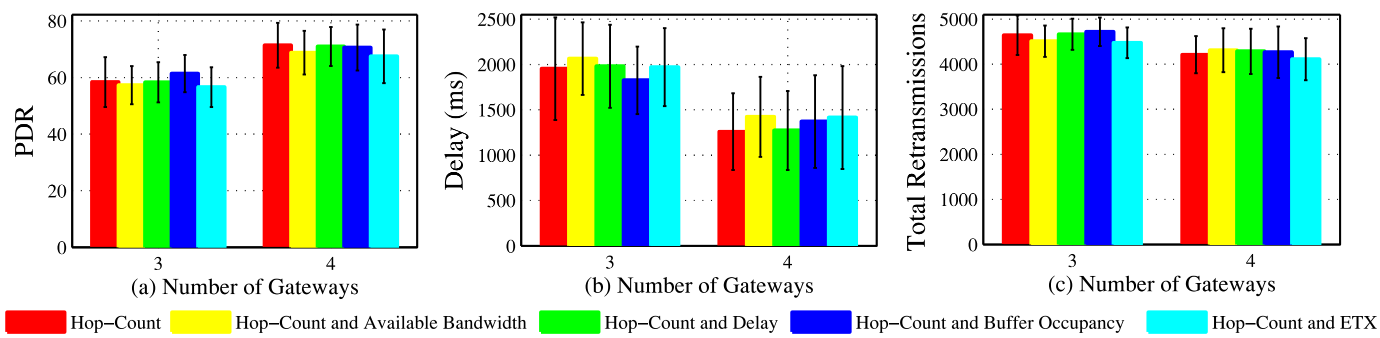

Fig. 7. RPL-based protocol performance using an end-to-end approach and increased data generation (Null RDC).

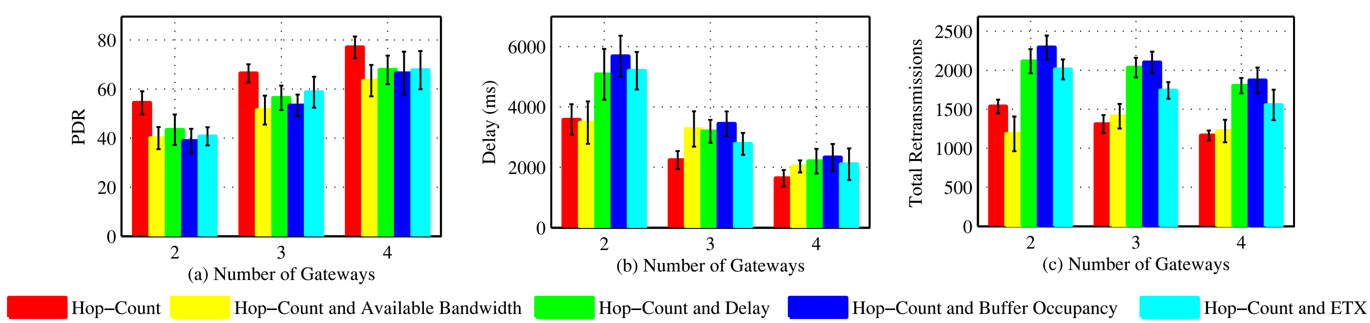

Fig. 8. RPL-based protocol performance using a greedy approach (ContikiMAC).

layed updates may result in selecting sub-optimal nodes along a data forwarding path.

(2) Fig. 10 shows that the total data packets dropped by the other protocols is statistically significantly higher than the number of packets dropped by the hop-count-based routing protocol. Comparison of the other protocols reveal that they all drop a similar number of packets. The hop-count-based routing protocol only uses a fixed downstream node, but the other protocols switch downstream nodes based on the tie-breaking metrics' value. Therefore, data transmission synchronization with multiple nodes under the duty cycling is hard, and it is negatively impacting the performance of those protocols that switch their downstream nodes.

The same set of simulation-based experiments are repeated by using the tie-breaking metric on the end-to-end basis, and Fig. 9 shows the protocols' performance. Apart from the available-bandwidth-based protocol all other protocols demonstrate similar PDR, delay, and total retransmissions. The higher control overhead associated with the available-bandwidth-based protocol negatively impacts the protocol's PDR, and it also limits the total number of data packets that can be transmitted using the protocol. As the available-bandwidth-based protocol transmits a lower number of data packets compared to the other protocols, the protocol demonstrates lower total retransmissions. Comparing the results shown in Fig. 9 with the results shown in Fig. 8 reveals that, using the tie-breaking metric on an end-to-end basis is better compared to the greedy approach. This is different to the results that were obtained without using the duty cycling algorithm. Using the end-to-end approach, the routing protocol only changes a data forwarding path if a new path advertises a better value of a tie-breaking metric. In the duty cycling operation, the time required to propagate the tie-breaking metric's updated value to a source node is relatively high. Hence, nodes do not switch paths frequently. Infrequent switching of the paths results in transmission synchronization along the forwarding path, hence better performance.

Fig. 11 shows the total number of control messages transmitted in a simulation under the duty cycling operation. The results are consistent with the results shown in Fig. 4. The only difference is that, under duty cycling fewer control messages are transmitted compared to the number of control messages transmitted without duty cycling. The reason being, the probability of successful data transmission is lower under duty cycling, as a result packet/frame queues fill up, and packets are dropped.

Fig. 12 shows the total number of data packets transmitted by the protocols under the duty cycling operation. Apart from the available-bandwidth-based protocols all of the other protocols

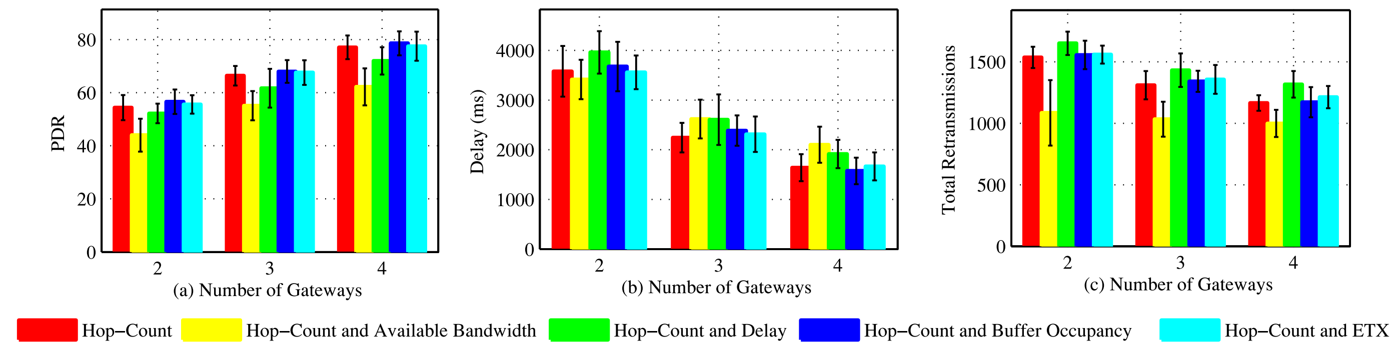

Fig. 9. RPL-based protocol performance using an end-to-end approach (ContikiMAC).

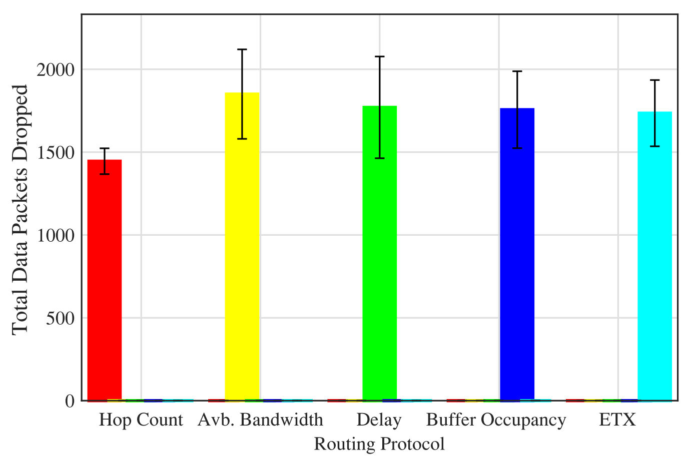

Fig. 10. Packet drop comparison (ContikiMAC).

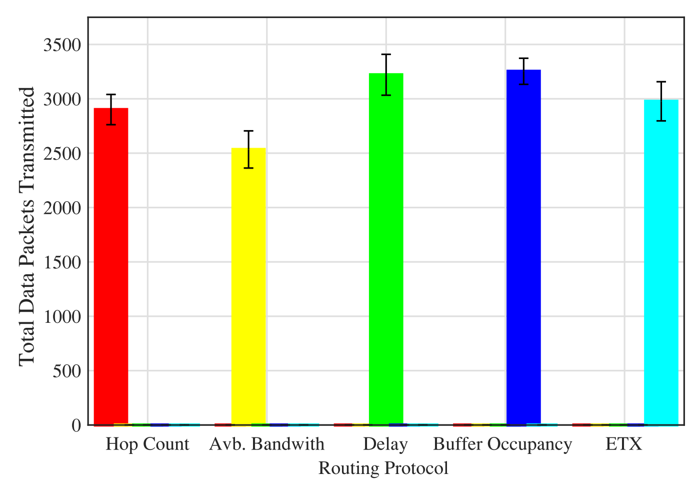

Fig. 12. Total data packets transmitted (ContikiMAC).

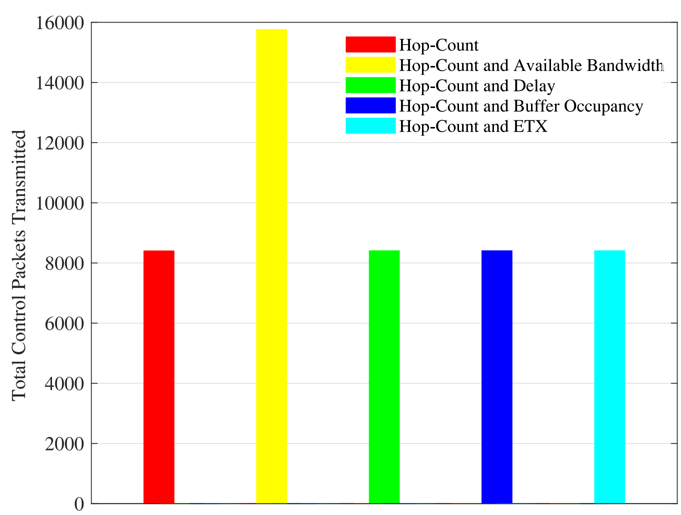

Fig. 11. Control overhead (ContikiMAC).

demonstrate a comparable number of data packet transmissions. Comparison of the results presented in Figs. 12 and 5 reveals that, apart from the available-bandwidth-based protocol, the other protocols transmitted fewer packets. The available-bandwidth-based protocol transmitted a higher number of packets due to the following reasons:

(1) The control packets to data packets ratio is higher in the available-bandwidth-based protocol. The control packets are multicast, hence do not require an acknowledgement. Due to the stated reason, the probability of a data packet being dropped due to full MAC layer outgoing queue is lower compared to the case when there are considerable unicast packets in the queue that require acknowledgements.
(2) In duty-cycling operation, the number of control packets transmitted by the available-bandwidth-based protocol is lower.
(3) For available-bandwidth-based routing (1) and (2) resulted in a higher number of data packets transmission by the available-bandwidth-based protocols compared to the data packet transmitted by the same protocols with no duty-cycling.

We performed another set of simulations by changing the packet generation distribution as the number of gateway nodes increases. For 3 and 4 gateways, the distribution changes to [2, 5] packets/sec and [2, 6] packets/sec respectively. The same set of experiments with the same distribution were also carried out without duty cycling. Fig. 13 demonstrates the routing protocols' performance with the increased data generation rate and using the greedy approach. The hop-count-based protocol again demonstrates statistically significantly higher PDR, lower delay, and lower retransmissions compared to the other protocols. However, the protocol demonstrates a similar number of retransmissions compared to the available-bandwidth-based protocol. This is because, the available-bandwidth-based protocol transmits fewer data packets, hence it demonstrates a similar number of retransmissions. The reasons for the better performance of the hop-count-based protocol are the same as discussed for the case when the packet generation distribution does not change as the number of gateways increases. Comparison of Fig. 13 with Fig. 6 reveals that, without duty cycling, the protocols demonstrate similar performance. However, in duty cycling operation the protocols that also use a tie-breaking metric demonstrate poor performance. Again, switch-

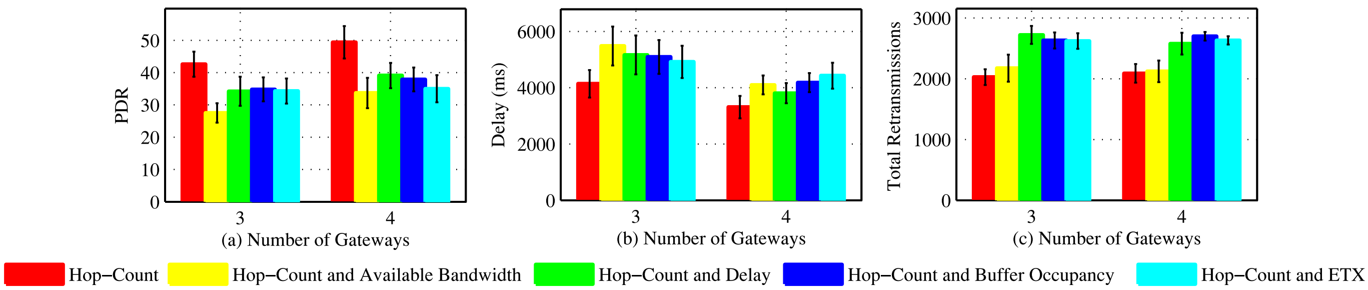

Fig. 13. RPL-based protocol performance using a greedy approach and increased data generation (ContikiMAC).

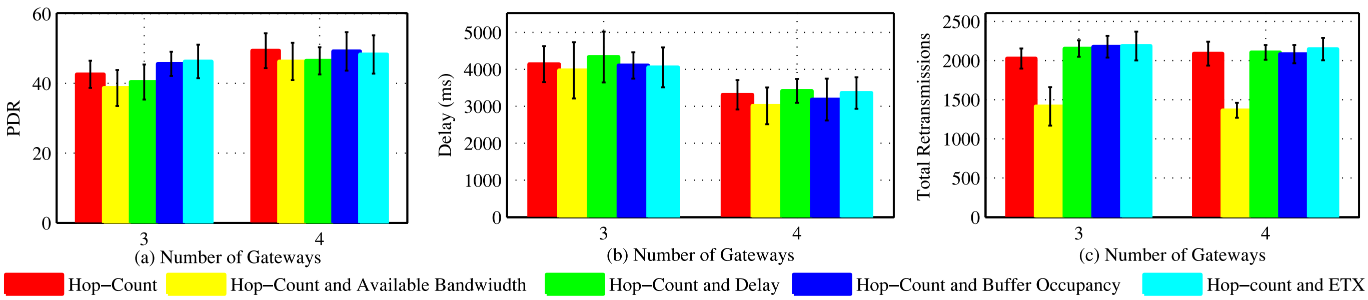

Fig. 14. RPL-based protocol performance using an end-to-end approach and increased data generation (ContikiMAC).

ing downstream nodes results in the transmission synchronization problem with multiple nodes, hence the problem negatively impacts such protocols' performance.

Fig. 14 demonstrates the routing protocols' performance with the increased data generation rate and using the end-to-end approach. The protocols demonstrate similar PDR, delay, and retransmissions. However, the available-bandwidth-based protocol demonstrates statistically significantly lower retransmissions. The reduced number of retransmissions is due to the fact that the protocol transmits fewer data packets due to the higher control overhead. Using the greedy approach, the hop-count-based protocol demonstrated better performance, but this not true for the results presented in Fig. 14. Again, infrequent path switching improves the performance of the other protocols. Mostly, the results presented in Fig. 14 are consistent with the results presented in Fig. 7. However, in this case, the mean PDR is lower and delay is higher due to the duty cycling operation.

5.3. Discussion

The research presented in this paper highlights the following key points:

1. RPL OFs Performance in the Absence of Duty-Cycling
   (a) Constant Data Load Across Different Number of Gateways: Greedy OFs perform better compared to the end-to-end OFs. In end-to-end OFs, if the value of the tie-breaking metric deteriorates multiple hops away from the source, some time is required to propagate the changed value to the source. Therefore, for some time sub-optimal paths are used. Overall, the greedy OFs that use delay, buffer occupancy, and ETX as a tie-breaking metric in conjunction with the shortest hop-count metric are better.
   (b) Increased Data Load With an Increase in the Number of Gateways: In a relatively high data traffic load, all OFs demonstrate similar and higher number of retransmissions. This is an indication of congestion. In the congested network, all OFs perform similarly.
2. RPL OFs Performance Using Duty-Cycling
   (a) Constant Data Load Across Different Number of Gateways: In duty-cycling operation, synchronization among nodes is an issue. Frequently switching parent nodes requires a node to synchronize with different nodes. This requires extra time, hence successful packet delivery is negatively impacted. As OFs based on the end-to-end approach result in infrequent parent changes compared to the OFs based on greedy approach, the end-to-end approach is better. Using the hop-count routing metric, nodes rarely change their parent, hence the hop-count metric demonstrate better performance.
   (b) Increased Data Load With an Increase in the Number of Gateways: In a relatively high data traffic load, the different routing metrics perform similarly. The OFs based on the end-to-end approach perform better compared to the OFs based on the greedy approach. The reason for this is again reduced switching of parent nodes. Overall, in duty-cycling operation, a simple OF can perform well, hence no need for more complicated OF.
3. Use of Tie-Breaking Metrics and Single Gateway
   (a) Tie-Breaking Metrics: Mostly, LLNs have redundancy in terms of the available forwarding paths to a gateway. Therefore, using a single shortest hop-count-based forwarding path can result in congestion on the path, and it is also an inefficient use of the available resources. Moreover, LLNs' use cases can have particular requirements in terms of reliability, delay, and bandwidth, therefore exploiting the redundancy in terms of the available shortest paths and using the tie-breaking metrics can help to select a path that better suits the requirements of a particular LLN use case.
   (b) Single Gateway: The work presented in this paper was performed on the assumption that a single gateway in LLN can result in congestion, hence the single gateway may not suit many LLNs' use cases. Our results demonstrated that, as we increase the number of gateways in the network, the network showed improved performance. Hence, our results confirm our assumption.

6. Conclusions and future work

We designed and analysed multiple OFs for multi-gateway ad-hoc LLNs. The OFs are used to discover and maintain data forwarding paths based on the requirements of LLNs. The OFs use available bandwidth, delay, buffer occupancy, and ETX as a tie-breaking metric in conjunction with the shortest hop-count metric. One set of OFs uses the tie-breaking metrics on a greedy basis and the other set uses them on an end-to-end basis. We analysed the impact of duty-cycling, the number of gateways, and the data traffic load in a multi-gateway ad-hoc LLN on the performance of our different OFs. In the absence of duty-cycling, the OFs that use the tie-breaking metrics on a greedy basis demonstrated the best performance. Among the analysed metrics, delay, MAC layer buffer occupancy, and ETX in conjunction with the shortest hop-count performed the best. In case of congestion, the presented OFs demonstrated similar performance. In duty-cycling mode, frequently switching parent nodes results in an extra synchronization time between a sender and receiver, hence it negatively impacts the performance. In an end-to-end approach, nodes infrequently switch their parent compared to the greedy approach, hence the end-to-end approach demonstrated the better performance.

The following are the main conclusions of our research: (*i*) in the absence of duty cycling, we should use the greedy routing approach for better performance, and delay, buffer occupancy, and ETX metrics should be used as the tie-breaking metrics, and (*ii*) in duty-cycling mode, the shortest hop-count metric should be used, and if there is a need to use the tie-breaking metrics they should be used in the end-to-end manner. We anticipate that our OFs can yield more benefit in more dense networks because more candidate forwarding paths will exist. Therefore, in the future we plan to evaluate the OFs in more dense networks.

Industrial IoT, remote health monitoring, assisted living, sharing multimedia content in a home/office environment, and smart homes are some of the many use cases for LLNs. Among the mentioned use cases some require reliability and other require bounded delay and/or minimum bandwidth guarantee. The presented OFs cannot only help to satisfy the requirements of the different LLNs' use cases, but our analysis can also help network engineers to choose an appropriate OF keeping in view the requirements of a particular LLN use case. In future, we also plan to analyse the presented OFs for downward and peer-to-peer communications.

References

[1] M.O. Farooq, T. Kunz, Wireless sensor networks testbeds and state-of-the-art multimedia sensor nodes, Appl. Math. Inf. Sci. 8 (3) (2014) 935–940.
[2] T. Arampatzis, J. Lygeros, S. Manesis, A survey of applications of wireless sensors and wireless sensor networks, in: IEEE International Symposium on Intelligent Control, 2005, pp. 719–724.
[3] H. Yoo, M. Shim, D. Kim, K.H. Kim, GLOBAL: agradient-based routing protocol for load-balancing in large-scale wireless sensor networks with multiple sinks, in: IEEE Symposium on Computers and Communications, 2010, pp. 556–562.
[4] L. Mottola, G.P. Picco, MUSTER: adaptive energy-aware multisink routing in wireless sensor networks, IEEE Trans. Mob. Comput. 10 (12) (2011) 1694–1709.
[5] H. Yoo, M. Shim, D. Kim, K.H. Kim, A scalable multisSink gradient-Based routing protocol for traffic load-Balancing, Eurasip J. Wireless Commun.Networking 2011 (85) (2011) 1–16.
[6] A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, J. Vasseur, R. Alexander, RPL: IPv6 Rotuing protocol for low-power and lossy networks, 2012, Last accessed on 29th May, 2015, [Online]. Available: https://tools.ietf.org/html/rfc6550.
[7] P. Ciciriello, L. Mottola, G. Picco, Efficient routing from multiple sources to multiple sinks in wireless sensor networks, in: Lecture Notes in Computer Science, 4373, 2007, pp. 34–50.
[8] A. Cabrera, A.J. Yuste-Delgado, D. Macías, A fuzzy logic-based and distributed gateway selection for wireless sensor networks, in: Highlights in Practical Applications of Agents and Multiagent Systems, 89, 2011, pp. 243–248.
[9] S. Isik, M.Y. Donmez, C. Ersoy, Multi-sink load balanced forwarding with a multi-Criteria fuzzy sink selection for video sensor networks, Comput. Netw. 56 (2) (2012) 615–627.
[10] J. Feng, L. Zheng, J. Fu, Z. Liu, An optimum gateway discovery and selection mechanism in WSN and mobile cellular network integration, in: 8th ICST International Conference on Communications and Networking in China, 2013, pp. 483–487.
[11] D. Kominami, M. Sugano, M. Murata, T. Hatauchi, Controlled potential-based routing for large-scale wireless sensor networks, in: Proceedings of the 14th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems, 2011, pp. 187–196.
[12] H. Jiang, R. Sun, Energy optimized routing algorithm in multi-sink WSNs, Appl. Math. Inf. Sci. 8 (1) (2014) 349–354.
[13] A. Boukerche, A. Martirosyan, An energy efficient and low latency multiple events' propagation protocol for wireless sensor networks with multiple sinks, in: Proceedings of the 4th ACM International Workshop on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks, 2007, pp. 82–86.
[14] M. Paone, L. Paladina, M. Scarpa, A. Puliafito, A multi-sink swarm-based routing protocol for wireless sensor networks, in: IEEE Symposium on Computers and Communications, 2009, pp. 28–33.
[15] N.Q. Dinh, T.D. Hoa, D.S. Kim, Distributed traffic aware routing with multiple sinks in wireless sensor networks, in: 9th IEEE International Conference on Industrial Informatics, 2011, pp. 404–409.
[16] D. Tan, D.S. Kim, Dynamic traffic-aware routing algorithm for multi-sink wireless sensor networks, Wireless Netw. 20 (6) (2014) 1239–1250.
[17] J. Guo, P. Orlik, J. Zhang, K. Ishibashi, Reliable routing in large scale wireless sensor networks, in: 6th International Conference on Ubiquitous and Future Networks, 2014, pp. 99–104.
[18] N. Mitton, D. Simplot-Ryl, M.E. Voge, L. Zhang, Energy efficient K-Anycast routing in multi-sink wireless networks with guaranteed delivery, in: Ad-hoc, Mobile, and Wireless Networks, 7363, 2012, pp. 385–398.
[19] F. Gara, L.B. Saad, R.B. Ayed, B. Tourancheau, RPL protocol adapted for heath care and medical applications, in: International Wireless Communication and Mobile Computing Conference, 2015, pp. 690–695.
[20] O. Gaddour, A. Koubäa, R. Rangarajan, O. Cheikhrouhou, E. Tovar, M. Abid, Co-RPL: RPL routing for mobile low power wireless sensor networks using corona mechanism, in: 9th IEEE International Symposium on Industrial Embedded Systems, 2014, pp. 200–209.
[21] K. Andrea, R. Simon, Design and evaluation of an RPL-based routing protocol for low-poer and lossy networks, in: 18th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile System, 2015, pp. 141–150.
[22] J. Guo, P. Orlik, K. Parsons, K. Ishibashi, D. Takita, Resource aware routing protocol in heterogeneous wireless machine-to-machine networks, in: Global Communications Conference, 2015, pp. 1–6.
[23] M. Zhao, I. Ho, P.H.J. Chong, An energy-efficient region-based RPL routing protocol for low-Power and lossy networks, IEEE Internet Things J. PP (99) (2016). 1–1
[24] K. Iwanicki, RNFD: routing-layer detection of DODAG (root) node failures in low-power wireless networks, in: Proceedings of the 15th International Conference on Information Processing in Sensor Networks, 2016, pp. 13:1–13:12.
[25] M.O. Farooq, C.J. Sreenan, K. Brown, T. Kunz, RPL-based routing protocols for multi-sink wireless sensor networks, in: 11th IEEE International Conference on Wireless and Mobile Computing, Network and Communication, 2015, pp. 452–459.
[26] K. Heurtefeux, H. Menouar, Experimental evaluation of a routing protocol for wireless sensor networks: RPL under study, in: 6th Joint IFIP Wireless and Mobile Networking Conference, 2013, pp. 1–4.
[27] J. Tripathi, J.C. de Oliveira, J. Vasseur, A performance evaluation study of RPL: routing protocol for low power and lossy networks, in: 44th Annual Conference on Information Sciences and Systems, 2010, pp. 1–6.
[28] O. Gaddour, A. Koubaa, S. Chaudhry, M. Tezeghdanti, R. Chaari, M. Abid, Simulation and performance evaluation of DAG construction with RPL, in: 3rd International Conference on Communications and Networking, 2012, pp. 1–8.
[29] D. Carels, N. Derdaele, E.D. Poorter, W. Vandenberghe, I. Moerman, P. Demeester, Support of multiple sinks via a virtual root for the RPL routing protocol, EURASIP J. Wireless Commun.Networking 2014 (91) (2014) 1–23.
[30] M.O. Farooq, T. Kunz, Proactive bandwidth estimation for IEEE 802.15.4-based networks, in: IEEE 77th Vehicular Technology Conference, 2013, pp. 1–5.
[31] M.O. Farooq, T. Kunz, BandEst: measurement-based available bandwidth estimation and flow admission control algorithm for Ad Hoc IEEE 802.15.4-based wireless multimedia networks, Int. J. Distrib. Sens. Netw. 2015 (2015) 1–15. Article ID 539048
[32] F. Osterlind, A. Dunkels, J. Eriksson, N. Finne, T. Voigt, Cross-level sensor network simulation with Cooja, in: 31st IEEE Conference on Local Computer Networks, 2006, pp. 641–648.
[33] M.O. Farooq, T. Kunz, Contiki-based IEEE 802.15.4 channel capacity estimation and suitability of its CSMA-CA MAC layer protocol for real-time multimedia applications, Mob. Inf. Syst. 2015 (2015) 1–15. 398637
[34] A. Dunkels, The ContikiMAC Radio Duty Cycling Protocol, Technical Report, Swedish Institute of Computer Science, 2011.

Muhammad Omer Farooq is a Senior Research Scientist in Nimbus Centre for Embedded Systems Research, Cork, Ireland. He was a Post-Doctoral Researcher in Computer Science Department at University College Cork in Ireland. He received Ph.D. degree in Computer Engineer from University of Luebeck, Germany. He has published more than 30 technical papers in International journals and conferences. His research interests are in designing protocols for the Internet of Things, interoperability among smart home communication protocols and technologies, routing protocols, heterogeneous networks, and quality of service in wireless networks. He is currently on the editorial boards of the IEEE Access and the KSII Transactions on Internet and Information Systems.

Cormac J. Sreenan received the Ph.D. degree in Computer Science from Cambridge University. He is a Professor of Computer Science at University College Cork in Ireland. Prior to joining UCC in 1999, he was on the research staff at AT&T Labs Research, Florham Park, New Jersey, and at Bell Labs, Murray Hill, New Jersey. He is currently on the editorial boards of the IEEE Transactions on Mobile Computing, the ACM Transactions on Sensor Networks, and the ACM/Springer Multimedia Systems Journal. He is a fellow of the British Computer Society and a member of the IEEE and the ACM.

Kenneth N. Brown is a Professor of Computer Science at University College Cork in Ireland, and the Deputy Director of 4C: Cork Constraint Computation Centre. Prior to that he was a senior lecturer at the University College Cork, lecturer at the University of Aberdeen, a Research Fellow at Carnegie Mellon University, and a Research Associate at the University of Bristol. His research interests are in the application of AI, optimisation and distributed reasoning, with a particular focus on wireless networks.

Thomas Kunz is a Professor in Computer and Systems Engineering at Carleton University Canada. He heads the Mobile Computing Group, researching wireless network architectures (manets, wireless mesh networks, wireless sensor networks, and smart cities), network protocols (routing, Mobile IP, QoS), and middleware layer for innovative wireless applications. He is author or co-author of more than 200 technical papers and received number of best paper awards.