CORA Cork Open Research Archive
Cartlann Taighde Oscailte Chorcaí

UCC Library and UCC researchers have made this item openly available. Please let us know how this has helped you. Thanks!

Title	RPL-based routing protocols for multi-sink wireless sensor networks
Author(s)	Farooq, Muhammad O.; Sreenan, Cormac J.; Brown, Kenneth N.; Kunz, Thomas
Publication date	2015-10
Original citation	Farooq, M. O., Sreenan, C. J., Brown, K. N. and Kunz, T. (2015) 'RPL-based routing protocols for multi-sink wireless sensor networks'. 2015 IEEE 11th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), 19-21 Oct , Abu Dhabi, United Arab Emirates, pp. 452-459. doi: 10.1109/WiMOB.2015.7347997
Type of publication	Article (peer-reviewed) Conference item
Link to publisher's version	http://dx.doi.org/10.1109/WiMOB.2015.7347997 Access to the full text of the published version may require a subscription.
Rights	**© 2015 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.**
Item downloaded from	http://hdl.handle.net/10468/5073

Downloaded on 2020-12-02T04:37:11Z

UCC
University College Cork, Ireland
Coláiste na hOllscoile Corcaigh

RPL-Based Routing Protocols For Multi-Sink Wireless Sensor Networks

Muhammad Omer Farooq *, Cormac J. Sreenan *, Kenneth N. Brown *, and Thomas Kunz †
* CTVR, Department of Computer Science, University College Cork, Ireland
† Department of Systems and Computer Engineering, Carleton University, Ottawa, Canada
Email: omer.farooq@insight-centre.org, cjs@cs.ucc.ie, k.brown@cs.ucc.ie, tkunz@sce.carleton.ca

***Abstract*—Recent studies demonstrate that the performance of a wireless sensor network (WSN) can be improved by deploying multiple sinks in the network. Therefore, in this paper we present different routing protocols for multi-sink WSNs based on the routing protocol for low-power and lossy networks (RPL). Our protocols use different routing metrics and objective functions (OFs). We use the available bandwidth, delay, MAC layer queue occupancy, and expected transmission count (ETX) as the tie-breaking metrics in conjunction with the shortest hop-count metric. Our OFs use the tie-breaking metrics on a greedy or end-to-end basis. Our simulation results demonstrate that the protocols based on the delay, buffer occupancy, and ETX metrics demonstrate best performance, increasing the packet delivery ratio by up to 25% and decreasing the number of retransmissions by up to 65%, compared to a version of the RPL protocol that only uses the hop-count metric. Another key insight is that, using the tie-breaking metrics on a greedy basis demonstrates a slight performance improvement compared to using the metrics on an end-to-end basis. Finally, our results also demonstrate that multiple sinks inside a WSN improve the RPL-based protocol performance.**

***Index Terms*—RPL, Multi-Sink WSN, Routing Protocols, Low-Power and Lossy Networks, IEEE 802.15.4.**

I. Introduction

A wireless sensor network (WSN) is composed of wireless sensor nodes and a sink node. The nodes are wirelessly interconnected with each other and with the sink. Such networks are characterized as low-power and lossy networks (LLNs) because nodes possess limited power and they operate in harsh environments. There are many applications of WSNs, 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 sink. If a node is not in a direct communication range of the sink, the data is reported in a multi-hop manner. Therefore, nodes closer to the sink relay data of those nodes that are further from the sink, hence hotspots can occur near the sink. These hotspot nodes tend to deplete the energy faster, which reduces WSN lifetime. Recent studies demonstrate that using multiple sinks inside a WSN can improve the network's performance and lifetime [3], [4], [5].

This work is supported by the grant (SFI 10/CE/I 1853) from Science Foundation Ireland as part of CTVR.

Depending on the application, data generated by nodes can have different end-to-end packet delivery delay and reliability requirements. For example, a WSN 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 sinks, therefore the routing protocol plays a pivotal role in delivering data to the sink. 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 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 LLN, limitations and satisfy heterogeneous application requirements.

We present multiple RPL-based routing protocols for multi-sink WSNs. Our protocols use the available bandwidth, delay, MAC layer buffer occupancy (the number of frames in the MAC layer queue), and expected transmission count (ETX) as the tie-breaking routing metrics in conjunction with the shortest hop-count metric. Our protocols' OFs can use the metrics on either a greedy or an end-to-end basis. Performance evaluations demonstrate that the RPL-based protocols designed using the delay, buffer occupancy, and ETX metrics perform the best, as they increase the packet delivery ratio (PDR) by up to 25% and decrease the number of retransmissions by up to 65%, compared to a version of RPL that only uses the hop-count metric. Another key insight is that, using the different tie-breaking metrics in a greedy manner shows a slight performance improvement compared to using them on an end-to-end basis. In general, the relative performance of the protocols is consistent as we increase the number of sinks and data traffic in a network. This is the first research paper that evaluates RPL-based protocols using different routing metrics and OFs in multi-sink WSNs.

The rest of this paper is organised as follows: a description of RPL is presented in Section II, related work is presented in Section III, our RPL-based routing protocols for multi-sink WSNs are presented in Section IV, simulation results are presented in Section V, and finally our conclusions and future work are given in Section VI.

II. 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 sink[1]), point to multi-point (sink 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 sink. 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 sink to nodes and from nodes to nodes, it contains prefix reachability information. As this paper focuses on upward communication[2], therefore the rest of discussion in this section is about upward route construction and maintenance in RPL.

A. DODAG Construction

Initially, the sink broadcasts the DIO message. Nodes in the transmission range of the sink receive the DIO message and decide to join the DODAG based on their OF. If the nodes join the DODAG, the nodes periodically broadcast the message. If the node does not join the DODAG, it only broadcasts the message. The process repeats at each node, and allows nodes to select their parent nodes towards the sink. Leaf nodes only join the DODAG, but do not broadcast 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 other that offers least delay. A single DODAG is termed as 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.

B. Routing Metrics and Constraints Support

Because of the diverse applications of LLNs and their energy, processing, size, and memory limitations, it is very difficult to fix a single or a combination of routing metrics. 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.

C. 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 sink. 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 sink. In the greedy rule, a node can not 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.

D. 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 sink, therefore it has additional control messages overhead. The sink can initiate the global repair, if it receives an inconsistent identifier for the DIO message.

E. Frequency of DIO Messages

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

III. Related Work

Fig. 1 shows different categories of routing protocols for multi-sink 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 sinks. 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 sink, number of different source-sink flows passing through the neighbour, and number of sinks that can be served by the neighbour. Based on the values of the function, the minimum number of neighbours required to serve

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

[2]communication from nodes to the sink

all the sinks 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 Sink Selection. Routing protocols presented in [8], [9] use fuzzy algorithms to select the best sink 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 sink, number of one hop neighbours of the downstream node leading to the sink, remaining energy of the downstream nodes, distance of the downstream nodes to the sink, and buffer occupancy at downstream nodes. Nodes periodically advertise the information required by the algorithms. The algorithms are periodically executed at nodes considering all sinks. The best sink 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 Sink Selection. The routing protocols presented in [10], [11], [12], [13], [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, the gradient fields to all sinks 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 sink to which it has the steepest gradient. Relaying nodes forward the packet on a path that offers the steepest gradient to the sink. 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], [17] construct and maintain a best data forwarding path towards all sinks. This is done assuming an application selects the sink, hence a routing protocol does not select the sink node for a data packet (in general, this is the only difference compared to gradient-based best sink selection category). Gradient fields towards sinks are constructed using a combination of the following metrics: shortest hop-count, geographical distance, 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 sink selection category.

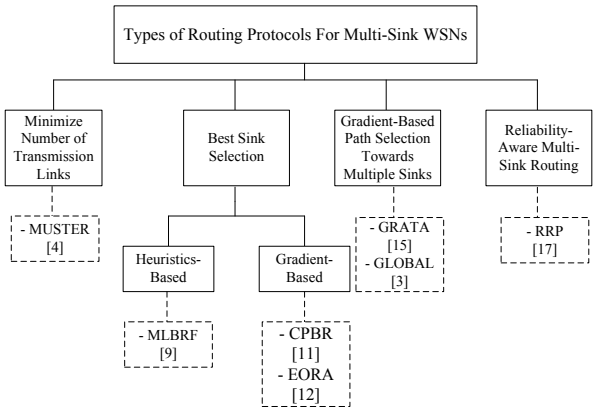

Fig. 1. Routing Protocols for Multi-Sink WSNs

Reliability-Aware Multi-Sink Routing. The routing protocols presented in [18], [19] aim to increase reliability. In [18], the routing protocol attempts to discover and maintain two disjoint data forwarding paths to each sink. Forwarding the same data packet on the two paths increases reliability as error probabilities on the paths are independent. In [19], the routing protocol constructs an energy-efficient minimum spanning tree towards K sinks among a total of M sinks in a network, and $K < M$. To increase reliability a data packet is forwarded to K sink nodes. Forwarding the same data packets on multiple paths incurs extra energy and can cause congestion in a network.

The state-of-the-art routing protocols for multi-sink WSNs do not evaluate different routing metrics' performance in multi-sink WSNs. Since RPL allows different metrics, we use it to evaluate the different metrics' performances. The performance of RPL itself has been evaluated in single-sink WSNs in [20], [21], [22] mostly using the hop-count and/or ETX metrics, and RPL is used for multi-sink WSN in [23], however there does not exist research work that evaluates RPL in a multi-sink WSN with different metrics and OFs. Hence, in this paper we present and evaluate different RPL-based routing protocols using different metrics and OFs in multi-sink WSNs.

IV. RRL-based Routing Protocols For Multi-Sink WSNs

In this section, we present different routing metrics, OFs, DODAG construction, sink selection, packet forwarding, and overheads of our routing protocols.

A. Routing Metrics

Our RPL-based routing protocols use the routing metrics related to throughput, delay, and reliability. We use the 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 the 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 [24], the algorithm has shown better results compared to the existing state-of-the-art. For the readers convenience, we briefly summarize this algorithm here. Using control messages, a node keeps track of data generation rates of nodes within the node's interference range. The IEEE 802.15.4's CSMA-CA MAC protocol also consumes bandwidth, e.g., a node can not transmit while it is in the 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 the wireless channel impairments (reflection, refraction, and multi-path fading) the algorithm uses sliding-window-based averaging to estimate the available bandwidth, and Equation 1 is used for available bandwidth estimation.

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

In Equation 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 [25] 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 μ^{th} 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 spend 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 second. The algorithm uses the sliding-window-based averaging with a window size of 5 seconds 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 seconds 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 seconds to obtain mean ETX at a node.

B. Objective Functions

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 sinks using the shortest hop-count routing metric. In case of multiple shortest paths to the same sink, one path is randomly selected.
2) Objective function 2 (OF2). Discover and maintain data forwarding paths to sinks using the shortest hop-count routing metric. In case there are multiple such paths to the same sink, 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) is selected. In case of multiple such candidate parents, a parent is selected randomly. OF2 is based on a greedy approach.
3) Objective function 3 (OF3). Discover and maintain data forwarding paths to sinks using the shortest hop-count routing metric. In case there are multiple such paths to the same sink, the path on which a candidate parent node has advertised a better end-to-end (from candidate parent to sink) value of the tie-breaking metrics (available bandwidth, delay, MAC layer buffer occupancy, and ETX) is selected. If there are more than one such candidate parents, a parent is randomly selected. OF3 is based on an end-to-end approach.

C. DODAG Construction, Sink Selection, and Data Forwarding

For DODAG construction DIO messages are used in the same way as described in Section II-A. 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 (sink node network layer address). The $rank$ field of the message contains the hop-count to the sink. 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 sinks 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 sinks: sink 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 sink 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 the 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 sinks, $s_{\text{i-rank}}$ 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 that RPL instance can take using 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 DODAG to a new sink or the advertised DODAG is better than the existing DODAG. Afterwards, the node updates its routing table, if required.

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 sink. 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. In Algorithm 1, $dio.s_{i-tie}$ contains the best value of the tie-breaking metric either on the end-to-end basis or on the greedy basis depending upon the routing protocol being used. The $join\_DODAG$ function also disassociates a node from any previously joined sub-optimal DODAG (if any) corresponding to the same $rpl_{instance}$.

Algorithm 1: DODAGs Construction and Maintenance

```
1  Input: dio;
2  routingRecord rt-rec;
3  i ← 0;
4  node-interested ← req-to-join ← false;
5  node-interested = is_interested(inst_n, dio.rpl_instance);
   rt-rec = search_rt_table(dio.rpl_instance, dio.s_i-id);
6  if rt-rec == NULL then
7      insert_rec_in_rt_table(dio);
8      if node-interested then
9          join_DODAG(dio);
10     end
11 end
12 else
13     if rt-rec.s_i-rank > dio.s_i-rank then
14         rt-rec.s_i-rank ← dio.s_i-rank;
15         rt-rec.s_i-parent ← dio.src_addr;
16         rt-rec.s_i-tie ← dio.s_i-tie;
17         req-to-join ← true;
18     end
19     else
20         if (dio.rpl_instance > 0) && (dio.rpl_instance < 9)
           then
21             if rt-rec.s_i-rank == dio.s_i-rank then
22                 if is_better(dio.s_i-tie, rt-rec.s_i-tie) then
23                     rt-rec.s_i-parent ← dio.src_addr;
24                     rt-rec.s_i-tie ← dio.s_i-tie;
25                     req-to-join ← true;
26                 end
27             end
28         end
29     end
30     if (node-interested) && (req-to-join) then
31         join_DODAG(dio);
32     end
33 end
```

Each data packet is forwarded to a single sink, therefore before forwarding the packet, a source node selects the closest sink in terms of hop-count. If there are multiple such sinks, one is selected randomly.

D. 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 [24]. Equation 2 can be used to determine network-wide mean control bits overhead per unit time. In Equation 2, OH is the overhead, $\overline{T}$ 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 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.

$$OH = \begin{cases} \big(n \times (\overline{T} \times l)\big) + (n \times i) & \overline{T} \leq j \\ \big(n \times (\overline{T} \times l)\big) + \Big(\lceil \frac{\overline{T}}{j} \rceil \times (n \times i)\Big) & \overline{T} > j \end{cases} \quad (2)$$

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 contrary, 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.

V. Simulation Results

Simulations were performed using the widely used Cooja WSN simulator [26] that uses real programming code for a wireless sensor node. We used a grid network topology with 75 nodes placed in a $300 \times 300\, m^2$ area. Based on published work we vary the number of sinks from 2 to 4, and sinks 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 seconds. Our traffic generation model is a representation of a data traffic generated by a range of event-detection system, e.g., fire detection, target tracking, etc. Our results are based on 10 simulation runs (randomly placing sinks each time) for each number of sink 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. Where CIs overlap and means are not in the overlap region, we base our conclusions on the result of a t-test. Table I shows general simulation parameters.

TABLE I
General Simulation Parameters

Parameter	Value
MAC layer	IEEE 802.15.4 CSMA-CA
MAC layer reliability	Enabled
Radio duty cycling algorithm	No radio duty cycling
Radio model	Unit disk graph with distance loss
Channel rate	250 kbps
MAC layer queue size	20 frames
Node transmission range	50 meters
Node carrier sensing range	100 meters
Total frame size	127 bytes
Motes emulated	Tmote sky

Fig. 2 demonstrates 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 sinks 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 metrics 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 sinks increases, as shown in Fig. 2 (b) and Fig. 2 (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. But, the latter set of protocols demonstrate statistically significantly lower retransmissions as compared to the former set of protocols. In case of four sinks 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 change 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 metrics on the end-to-end basis. The results shown in Fig. 3 show similar patterns as those discussed in Fig. 2. But, in case of 2 sinks, the protocols based on delay and buffer occupancy metrics demonstrate approximately 25% higher

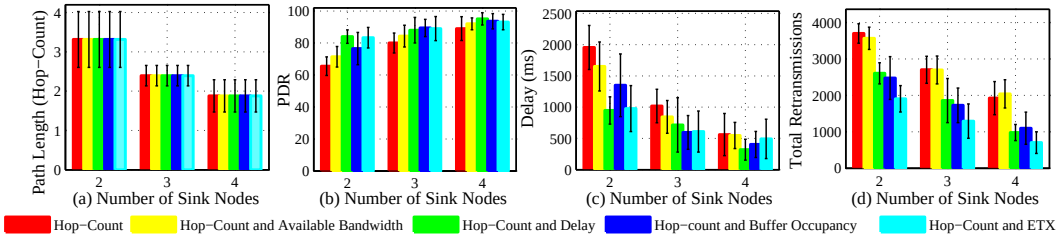

Fig. 2. RPL-based Protocol Performance Using a Greedy Approach

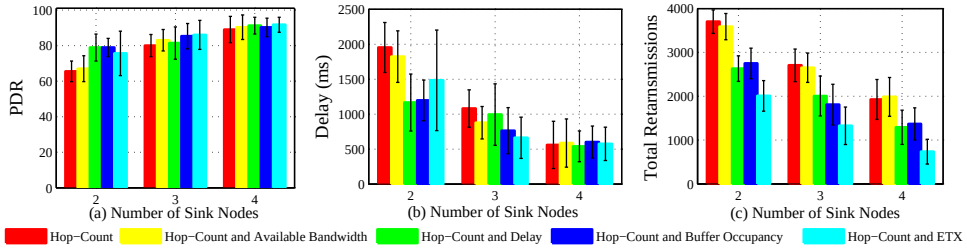

Fig. 3. RPL-based Protocol Performance Using an End-to-End Approach

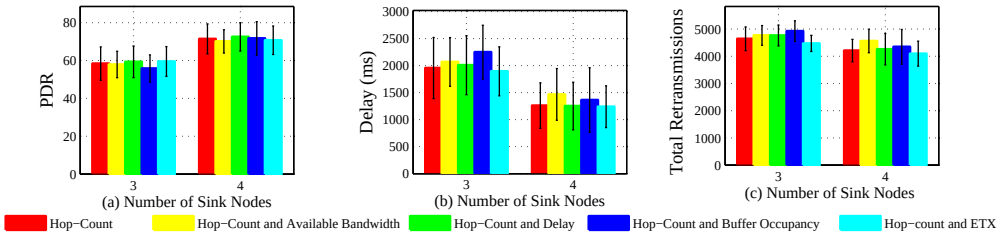

Fig. 4. RPL-based Protocol Performance Using a Greedy Approach and Increased Data Generation

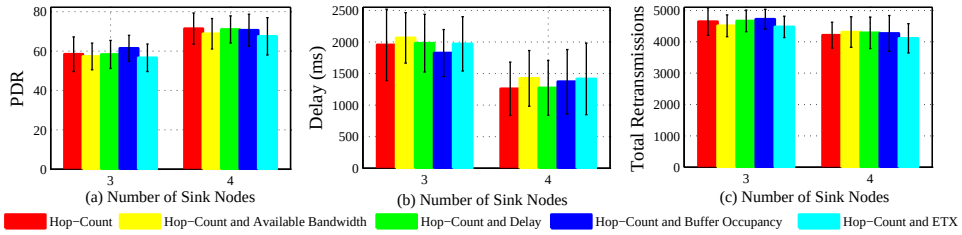

Fig. 5. RPL-based Protocol Performance Using an End-to-End Approach and Increased Data Generation

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, Fig. 4, and Fig. 5 do not plot mean path lengths as they are the same as shown in Fig. 2 (a). 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.

In the described set of simulations, we do not change the packet generation distribution as the number of sinks 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 sinks. For 3 and 4 sinks, the packet generation distributions change to [2, 5] packets/sec and [2, 6] packets/second respectively.

Fig. 4 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. 4 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. 5 shows the routing protocols' performance using the end-to-end approach and increased data generation rates. The protocols again demonstrate similar performance. Comparing the results presented in Fig. 4 and Fig. 5 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.

VI. Conclusions and Future Work

We designed and analysed multiple RPL-based routing protocols for multi-sink WSNs. The routing protocols used different routing metrics and objective functions. Among the presented and analysed protocols, those that used delay, MAC layer buffer occupancy, or ETX in conjunction with the shortest hop-count as routing metrics performed the best. The protocols based on the greedy approach performed slightly better than the protocols that use the end-to-end approach for data forwarding path selection. Therefore, the greedy approach is preferable as it has demonstrated better performance and it is easier to implement. As the number of sinks in a WSN increases, the protocols demonstrated better performance. But, in a state of congestion the protocols performed similarly. In the future we plan to design and evaluate more RPL-based protocols for multi-sink WSNs using other metrics, e.g., expected transmission time, and link quality indicator.

References

[1] M. O. Farooq and T. Kunz, "Wireless Sensor Networks Testbeds and State-of-the-Art Multimedia Sensor Nodes," *Applied Mathematics and Information Science*, vol. 8, no. 3, pp. 935–940, 2014.

[2] T. Arampatzis, J. Lygeros, and 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, and K. H. Kim, "GLOBAL: A Gradient-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 and G. P. Picco, "MUSTER: Adaptive Energy-Aware Multisink Routing in Wireless Sensor Networks," *IEEE Transactions on Mobile Computing*, vol. 10, no. 12, pp. 1694–1709, 2011.

[5] H. Yoo, M. Shim, D. Kim, and K. H. Kim, "A Scalable Multi-Sink Gradient-Based Routing Protocol For Traffic Load-Balancing ," *Eurasip Journal of Wireless Communication and Networking*, vol. 2011, no. 85, pp. 1–16, 2011.

[6] A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, J. Vasseur, and R. Alexander, "RPL: IPv6 Rotuing Protocol for Low-Power and Lossy Networks," RFC 6550, March 2012.

[7] P. Ciciriello, L. Mottola, and G. Picco, "Efficient Routing from Multiple Sources to Multiple Sinks in Wireless Sensor Networks," in *Lecture Notes in Computer Science*, 2007, vol. 4373, pp. 34–50.

[8] A. Cabrera, A. J. Yuste-Delgado, and D. Macas, "A Fuzzy Logic-Based and Distributed Gateway Selection for Wireless Sensor Networks," in *Highlights in Practical Applications of Agents and Multiagent Systems*, 2011, vol. 89, pp. 243–248.

[9] S. Isik, M. Y. Donmez, and C. Ersoy, "Multi-Sink Load Balanced Forwarding with a Multi-Criteria Fuzzy Sink Selection for Video Sensor Networks ," *Computer Networks*, vol. 56, no. 2, pp. 615 – 627, 2012.

[10] J. Feng, L. Zheng, J. Fu, and Z. Liu, "An Optimum Gateway Discovery and Selection Mechanism in WSN and Mobile cellular Network Integration," in 8^{th} *ICST International Conference on Communications and Networking in China*, 2013, pp. 483–487.

[11] D. Kominami, M. Sugano, M. Murata, and T. Hatauchi, "Controlled Potential-based Routing for Large-scale Wireless Sensor Networks," in *Proceedings of the* 14^{th} *ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, 2011, pp. 187–196.

[12] H. Jiang and R. Sun, "Energy Optimized Routing Algorithm in Multi-Sink WSNs," *Applied Mathematics & Information Sciences*, vol. 8, no. 1, pp. 349–354, 2014.

[13] A. Boukerche and A. Martirosyan, "An Energy Efficient and Low Latency Multiple Events' Propagation Protocol for Wireless Sensor Networks with Multiple Sinks," in *Proceedings of the* 4^{th} *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, and 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, and D. S. Kim, "Distributed Traffic Aware Routing with Multiple Sinks in Wireless Sensor Networks," in 9^{th} *IEEE International Conference on Industrial Informatics*, 2011, pp. 404–409.

[16] D. Tan and D. S. Kim, "Dynamic Traffic-Aware Routing Algorithm for Multi-Sink Wireless Sensor Networks," *Wireless Networks*, vol. 20, no. 6, pp. 1239–1250, 2014.

[17] D. Djenouri and I. Balasingham, "Traffic-Differentiation-Based Modular QoS Localized Routing for Wireless Sensor Networks," *IEEE Transactions on Mobile Computing*, vol. 10, pp. 797–809, 2011.

[18] J. Guo, P. Orlik, J. Zhang, and K. Ishibashi, "Reliable Routing in Large Scale Wireless Sensor Networks," in 6^{th} *International Conference on Ubiquitous and Future Networks*, 2014, pp. 99–104.

[19] N. Mitton, D. Simplot-Ryl, M. E. Voge, and L. Zhang, "Energy Efficient K-Anycast Routing in Multi-sink Wireless Networks with Guaranteed Delivery," in *Ad-hoc, Mobile, and Wireless Networks*, 2012, vol. 7363, pp. 385–398.

[20] K. Heurtefeux and H. Menouar, "Experimental Evaluation of a Routing Protocol for Wireless Sensor Networks: RPL Under Study," in 6^{th} *Joint IFIP Wireless and Mobile Networking Conference*, 2013, pp. 1–4.

[21] J. Tripathi, J. C. de Oliveira, and J. Vasseur, "A Performance Evaluation Study of RPL: Routing Protocol for Low power and Lossy Networks," in 44^{th} *Annual Conference on Information Sciences and Systems*, 2010, pp. 1–6.

[22] O. Gaddour, A. Koubaa, S. Chaudhry, M. Tezeghdanti, R. Chaari, and M. Abid, "Simulation and Performance Evaluation of DAG Construction with RPL," in 3^{rd} *International Conference on Communications and Networking*, 2012, pp. 1–8.

[23] D. Carels, N. Derdaele, E. D. Poorter, W. Vandenberghe, I. Moerman, and P. Demeester, "Support of Multiple Sinks Via a Virtual Root for the RPL Routing Protocol," *EURASIP Journal of Wireless Communications and Networking*, vol. 2014, no. 91, pp. 1–23, 2014.

[24] M. O. Farooq and T. Kunz, "Proactive Bandwidth Estimation for IEEE 802.15.4-Based Networks," in *IEEE* 77^{th} *Vehicular Technology Conference*, 2013, pp. 1–5.

[25] M. O. Farooq and T. Kunz, "BandEst: Measurement-Based Available Bandwidth Estimation and Flow Admission Control Algorithm for Ad Hoc IEEE 802.15.4-Based Wireless Multimedia Networks," *International Journal of Distributed Sensor Networks*, vol. 2015, pp. 1–15, 2015, article ID 539048.

[26] F. Osterlind, A. Dunkels, J. Eriksson, N. Finne, and T. Voigt, "Cross-Level Sensor Network Simulation with Cooja," in 31^{st} *IEEE conference on Local Computer Networks*, 2006, pp. 641–648.