ERPL: An Enhanced Peer-to-Peer Routing Mechanism for Low-Power and Lossy Networks

Muhammad Farooq, Dirk Pesch

HAL Id: hal-01995160

https://inria.hal.science/hal-01995160

Submitted on 25 Jan 2019

ERPL: An Enhanced Peer-to-Peer Routing Mechanism for Low-Power and Lossy Networks

Muhammad Omer Farooq and Dirk Pesch
Nimbus Research Centre
Cork Institute of Technology, Cork, Ireland
Email: omer.farooq@cit.ie, dirk.pesch@cit.ie

Abstract—The Routing Protocol for low-power and Lossy networks (RPL) is the most popular routing protocol for low-power and lossy networks (LLNs). Recent studies demonstrate that RPL performs poorly in peer-to-peer (P2P) communication. However, P2P communication is of immense importance in many LLNs that require actuation and control operations, such as cyber-physical systems. In order to alleviate the performance problem of RPL, we present a mechanism to enhance P2P route construction and data packet forwarding in RPL's storing and non-storing mode of operations (MoPs), which we call Enhanced RPL (ERPL). The salient features of ERPL include the following: (i) optimized P2P routing and data forwarding, (ii) no additional control messages, and (iii) ERPL can coexist with standard RPL implementations. We have implemented ERPL in the Contiki operating system and extensively evaluated it against a RPL implementation using Cooja-based emulation and physical testbed based experiments. Our results demonstrate that ERPL outperforms standard RPL in P2P communication and its optimized P2P route construction and data forwarding algorithms also positively impact the protocol's performance in multipoint to point (MP2P) and point to multipoint (P2MP) communications. Moreover, ERPL is more energy-efficient. Our results also shed light on the performance of MP2P, P2MP, and P2P communications relative to RPL's destination-oriented directed acyclic graph (DODAG) depth, i.e., a deeper DODAG negatively impacts the performance of MP2P and P2MP communications, however it positively impacts P2P communication, while the reverse holds true for a relatively shallow DODAG.

Index Terms—Low-Power and Lossy Networks (LLNs), Routing Protocols, DODAG Depth Analysis.

I. INTRODUCTION

Typically, Wireless Sensor Networks (WSNs), Internet of Things (IoT), and Cyber-Physical Systems (CPS) all require low-power and lossy networks (LLNs) for some or all of their operation. These networks are characterized as LLNs because nodes in such networks possess limited resources and usually operate in harsh communication environments, impacting the reliability of wireless links. Communication and computing devices are being increasingly embedded in objects and structures to enable networked sensing and actuation functions. This is resulting in sophisticated CPS, such as power generation and distribution networks, assisted living, traffic control and safety systems, autonomous vehicles, and distributed robotics. Unlike the pure sensing focus of traditional WSNs, CPS require distributed decision making, control and actuation. Hence, low power devices need to communicate not only with a control gateway, but need to communicate

also directly with other sensor or actuator devices. Here, we refer to these types of communication as multipoint to point (MP2P), point to multipoint (P2MP), and peer-to-peer (P2P) communications respectively.

Due to energy and radio communication limitations, devices in a LLN may not connect directly to the gateway or to other devices in the same network. Hence, devices need to collaborate to relay or multi-hop data packets wirelessly to each other, to the gateway and to other devices in the network. Hence, a routing protocol is required to discover MP2P, P2P, and P2MP data forwarding paths. Considering the limitations of LLNs, the Internet Engineering Task Force (IETF) ROLL (routing over low-power and lossy networks) working group has standardized the routing protocol for low-lower and lossy networks (RPL) [1]. RPL supports MP2P, P2MP, and P2P route discovery and data forwarding. Moreover, it is a generic routing framework for LLNs and uses the concept of an objective function (OF) to influence how routes are constructed in a particular LLN. If there is no RPL OF defined in the standard that meets the requirements of a particular LLN and/or to satisfy the requirements of application(s), a new OF can be designed. Hence, the concept of OFs adds flexibility, scalability, and adaptablity to RPL.

Despite RPL's flexibility, we highlight inefficiencies in its P2P route construction and data packet forwarding algorithms. Moreover, we present enhancements to improve P2P route construction and data forwarding in RPL's storing and non-storing mode of operations (MoPs) that we capture in an improved protocol, called Enhanced RPL (ERPL). The following are our main contributions:

- Identification of inefficiencies in P2P route construction and data forwarding in RPL.
- Improvements to P2P route construction and data forwarding algorithms for RPL's storing and non-storing MoPs.
- Analyzing the impact of RPL's destination-oriented directed acyclic graph (DODAG) depth on the MP2P, P2MP, and P2P communications.
- Our analysis of the protocols using different random and grid topologies by varying the data packet generation interval and changing the location of gateway in different networks. Our results that show that our ERPL protocol demonstrates overall lower packet loss, delay, it requires a fewer transmissions to deliver P2P packets, and it is more energy-efficient compared to RPL.

The remainder of this paper is organized as follows: RPL's brief introduction is presented in Section II. Related work is discussed in Section III, our enhanced P2P route construction and data forwarding algorithms for RPL are presented in Section IV. Performance evaluations are presented in Section V, and finally conclusions are presented in Section VI.

II. RPL: ROUTING PROTOCOL FOR LOW-POWER AND LOSSY NETWORKS

RPL [1] is a standard routing framework for LLNs. Generally, in a RPL network there is a border router also known as a gateway/root that connects different nodes in the network to an external network, such as the Internet. Within a network, RPL supports MP2P, P2MP, and P2P communications. RPL constructs a routing topology called destination-oriented directed acyclic graph (DODAG) rooted at the gateway. To discover data forwarding paths, RPL uses the following control messages:

- DIO: DODAG information object (used to discover forwarding paths for MP2P communication)
- DIS: destination information solicitation (a node multicasts this message to discover the route to the gateway)
- DAO: destination advertisement object (used to discover forwarding paths for M2MP and P2P communications)

A. DODAG Construction

Initially, the gateway/root multicasts the DIO message. Upon the reception of the DIO message, direct neighbors of the gateway decide to join the DODAG advertised in the DIO message based on their OF. If a node joins the DODAG, using the RPL trickle timer [1] the node also multicasts the DIO message. This process continues and eventually DODAG information reaches to all the nodes in a network. If any node in the network is not interested in joining the DODAG advertised in the DIO message, the node ignores the message. Multiple DODAGs can exist in the same network, and they are differentiated by their instance ID.

B. P2MP and P2P Route Construction

For P2MP and P2P communications RPL supports two MoPs: storing mode and non-storing mode. In the storing mode, a node stores and maintains a forwarding table containing forwarding paths to other nodes in its sub-DODAG. In the non-storing mode, a node does not store and maintain forwarding paths to the nodes in its sub-DODAG. In the non-storing mode, if a node wants to communicate with another node in a network, the node forwards its data packet to the gateway as the gateway stores and maintains forwarding paths to all the nodes in the network. For P2MP and P2P route constructions, RPL uses DAO message. Each node in a network unicasts the DAO message to its upstream/parent node towards the gateway. Based on the RPL's MoP, the parent node either relays the same DAO message to its parent after storing the route to the DAO message transmitting node, or it may add itself as a node on the path to the DAO message transmitting node, in the DAO message and transmits the modified DAO message to its parent. In this way, the gateway learns a data forwarding path to the node that initiated the DAO message transmission. If the nodes are working in the storing MoP, in the process they also learn the data forwarding path to the node that initiated the DAO message transmission.

III. RELATED WORK

In [2], the authors argue that interoperability among RPL's storing and non-string MoPs can result in a network that is more effective and fault-resilient. First, the authors demonstrate that coexistence of RPL's storing and non-storing MoPs in the same network can lead to different problems, such as network partitioning and routing loops. Afterwards, different enhancements to the RPL protocol are proposed. The modified protocol resolves the identified problems and facilitates interoperability between RPL's both MoPs in the same network. Another approach of coexistence of RPL's storing and non-storing MoPs is presented in [3]. The drawback of the stated approaches is that the MoP is configured statically, hence nodes cannot switch between MoPs dynamically. To overcome memory limitations of RPL's storing mode, a modification to the RPL protocol called D-RPL is proposed in [4]. D-RPL overcomes the limitation by replacing storing mode forwarding with multicast dissemination. If a node's memory limits are reached, D-RPL activates to overcome the limitation.

In [5], an analysis of the RPL protocol from the IoT perspective considering the following benchmarks is presented: reliability and robustness, mobility, resource heterogeneity, and scalability. After the analysis, the article concludes that to remain a viable option in the IoT domain, RPL needs improvement especially in P2P communications to support emerging IoT use cases. Experimental performance evaluation of RPL using hop-count and/or ETX routing metrics is presented in [6],[7], [8], and RPL performance evaluation for multi-gateway networks using different objective functions is presented in [9]. To reactively discover P2P communication paths, enhancements for RPL are proposed in [10] and [11]. However, the enhancements require additional control messages and substantial changes in the RPL routing protocol.

The existing work on RPL primarily focuses on the protocol's analyses in different scenarios, interoperability between storing and non-storing MoP, reducing memory requirement for the storing MoP, and reactive discovery of P2P routing paths. Existing work on reactive discovery of P2P paths requires additional control overhead, and typically this may not be a good approach because nodes in LLNs posses limited resources and bandwidth is also scarce. Therefore, the focus of this paper is to improve P2P routing and forwarding in RPL without using additional control messages and with minimal change to the protocol so as to maintain interoperability with standard RPL implementations. This aspect is largely missing in the literature. The goal of this effort is to better support emerging applications in LLNs especially those that require actuation.
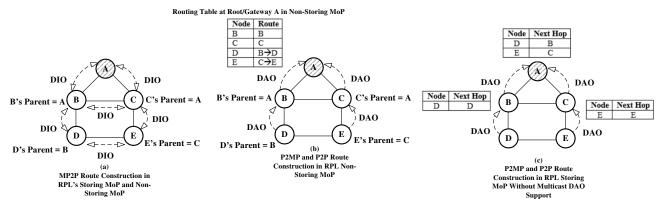
Fig. 1. RPL Operation with Storing and Non-Storing MoPs

IV. ENHANCED PEER-TO-PEER ROUTING IN LLNS

A. P2P Routing Problem in RPL

Fig. 1 shows RPL functionality in storing and non-storing MoPs. In the figure, we assume that the RPL OF 0 [12] is used. This objective function uses shortest hop-count routing metric. Fig. 1 (a) shows the construction of a RPL DODAG using DIO messages. Based on the OF, each node selects its preferred parent, and in the process MP2P routes are constructed. The MP2P route construction process is the same for storing and non-storing MoPs.

Fig. 1 (b) shows the P2MP and P2P route construction process in RPL's non-storing MoP. In this case, each node transmits a DAO message to its preferred parent node, and the parent node transmits the DAO message to its own preferred parent after appending its own information to the message. Eventually, the DAO message reaches the root of a DODAG. The root stores the complete route to each node. Whenever a node wants to communicate with a peer node, the node's data packet is first transmitted to the root, and the root uses source routing to forward the packet to the respective peer node. To illustrate this operation, let us consider the following scenario where node D wants to communicate with node E. In the non-storing MoP, nodes do not store a forwarding table, therefore D forwards the packet to the root (node A), and A then uses source routing to forward the packet to E. Therefore, the packet requires four transmissions $(D \Rightarrow B \Rightarrow A \Rightarrow C \Rightarrow E)$ despite the fact that D and E are direct neighbors. Moreover, the use of source routing requires additional bits to be transmitted by nodes along the path from the root to the destination. It is well-known that most of a node's energy is consumed during transmission and reception, hence in the given scenario RPL not only requires extra transmission to delivery a P2P data packet, however it also adds control overhead in data packet header due to source routing.

Fig. 1 (c) shows the P2MP and P2P route construction process in RPL's storing MoP. It is pertinent to mention here that Fig. 1 (c) shows the functionality of the existing RPL implementations that support storing MoP. In this MoP, a node advertises routes to the nodes in its sub-DODAG, for example, node D transmits a DAO message containing its reachability information to its parent (node B), and after storing a forward-

ing table entry for D, B transmits the same information in a DAO message to its parent (node A), hence in the process B and A learn a data forwarding path to D. Similarly, all other nodes in a network transmit DAO messages towards the root. For P2P forwarding, if a node has a forwarding entry for a destination node in its forwarding table, it forwards the packet to the next hop, otherwise the node forwards the packet to its parent node. This process continues until a common ancestor or root forwards the packet to the destination node. In this case, if D wants to communicate with E, the packet requires again four transmission $(D \Rightarrow B \Rightarrow A \Rightarrow C \Rightarrow E)$ despite the fact that D and E are direct neighbors. Hence, this forwarding algorithm is also not efficient. In the given scenario, it is also possible that node E wants to communicate with Node B. Using the RPL P2P data forwarding algorithm the packet follows the following path $(E \Rightarrow C \Rightarrow A \Rightarrow B)$, however B is the direct neighbor of node C, hence C can directly transmit the packet to B instead of relaying it to A, and this approach can reduce the number of transmissions required to deliver E's packet to B.

To eliminate the forwarding inefficiencies discussed above, the RPL standard proposes the use of multicast DAO (M_{DAO}) messages. Each node multicasts the message so that nodes in the network maintain forwarding table entries for their direct neighbors. Fig. 2 shows the P2MP and P2P route construction process in RPL's storing MoP with M_{DAO} messages. Due to space limitation, we are only show forwarding tables at root A and node D. In this case, if D wants to communicate with E, it directly forwards the packet to E. Hence, extra transmissions are avoided. However, this approach incurs extra control overhead as each node multicasts DAO messages apart from DIO and unicast DAO messages. Therefore, it is also not energy efficient.

B. Proposed Enhancement

We propose an enhancement for RPL's P2P route construction and data packet forwarding with the objective to avoid extra transmissions, if a destination node is a direct neighbor of a source node or the destination node is a direct neighbor of a node relaying a data packet. Our proposed enhancement fulfills the stated objective by avoiding RPL's M_{DAO} message, and at the same time no change is required in different RPL's control messages. Therefore, our proposed solution can easily
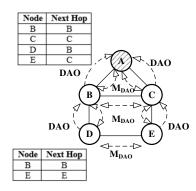
Node	Next Hop
B	B
C	C
D	B
E	C

Node	Next Hop
B	B
E	E

Fig. 2. P2MP and P2P Routing Using Multicast DAO

Node	Next Hop
B	B
C	C

Node	Next Hop
A	A
C	C
D	D

Node	Next Hop
A	A
B	B
E	E

Node	Next Hop
B	B
E	E

Node	Next Hop
C	C
D	C

(a)
Maintaining Forwarding
Information for Direct
Neighbours with DIO Message
Using ERPL

Node	Next Hop
B	B
C	C
D	B
E	C

Node	Next Hop
A	A
C	C
D	D

Node	Next Hop
A	A
B	B
E	E

Node	Next Hop
B	B
E	E

Node	Next Hop
C	C
D	C

(b)
P2MP and P2P Route
Construction with ERPL

Fig. 3. Operation of ERPL

coexist with existing standard RPL implementations. We call our protocol the enhanced routing protocol for low-power and lossy networks (ERPL).

RPL constructs MP2P routes using a DIO message. DIO messages corresponding to different DODAGs are multicasted in a network. If a node joins a DODAG advertised in a particular DIO message, it also multicasts the DIO message corresponding to that DODAG using RPL's trickle timer. If after processing a DIO message, a node decides not to join the DODAG advertised in the DIO message, the node ignores the DIO message. Therefore, in any case the node processes the DIO message. Due to the broadcast nature of the wireless channel, whenever a node multicasts a DIO message it is received by all of the node's direct neighbors. Using a DIO message, any node in a network can discover its direct neighbors and store forwarding table entries corresponding to all direct neighbors in its forwarding table. Therefore, we propose that instead of nodes using a M_{DAO} message to discover and maintain forwarding table entries for direct neighbors, the same can be achieved by using DIO messages and implementing the following modifications:

- Whenever a node receives a DIO message, regardless of the node's interest in joining the DODAG, the node extracts the mulitcasting node's address from the message and stores an entry for the mulitcasting node in its forwarding table, if the entry is not already present. Otherwise, the entry is refreshed. The next hop field for the forwarding table entry should be set equal to the mulitcasting node's address. Whenever a node has a packet to transfer, it consults the forwarding table using the destination address and if the entry is present, the packet is transmitted to the destination node or to the next hop. If an entry is not present, normal RPL forwarding rules are followed.

- Above works if a node operates in storing MoP, however in non-storing MoP nodes do not store a forwarding table. In this case, whenever a node receives a DIO message, it consults the neighbor table associated with the IPv6 neighbor discovery protocol. If the DIO message's multicasting node's address is not present in the neighbor table, a new record is created in the neighbor table and the multicasting node's address is stored in the table. In this case, the protocol stores a NULL value in the neighbor table's field corresponding to the link-
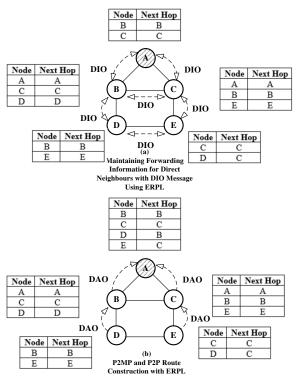
local address of a neighbor node. This forces the MAC layer to obtain a link-local address for the neighbor node before transmitting a frame. In non-storing MoP, another approach is to store a neighbor node's network layer address at the networking layer. This is a feasible approach in the non-storing MoP as nodes only need to store the network layer address of their direct neighbors. Usually, nodes have enough memory to store network layer addresses of their direct neighbors in a reasonably sized network. Whenever a node has a packet to transmit, using the destination node's address, the node consults its neighbor table or the data structure holding addresses of the node's direct neighbors. If a match is found, the packet is transfered to the destination node, otherwise RPL's standard forwarding algorithm for non-storing MoP is followed. When the root forwards the packet, it uses source routing, therefore the discussed forwarding algorithm is only valid when source routing is not used, i.e., when a P2P communication packet has not reached the root.

Fig. 3 (a) shows the ERPL functionality of maintaining direct neighbors' information using DIO messages. Similarly, Fig. 3 (b) shows P2MP and P2P route construction using ERPL. A comparison of Fig. 3 with Fig. 2 reveals that ERPL can achieve the same functionality with reduced control overhead. A comparison of Fig. 3 with Fig. 1 reveals that ERPL can achieve the same functionality with fewer transmissions and lower control overhead.

To get an idea of the benefits that the ERPL's P2P routing and data forwarding algorithms can provide, we performed computer-based simulations experiments. In our experiments,
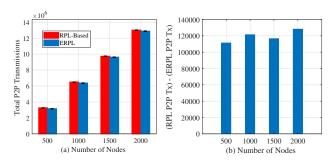
Fig. 4. P2P Transmissions Comparison

we vary the number of nodes in the network from 500 to 2000. In our network topologies, a node can have a maximum of 8 direct neighbors, and any node can reach the root in a maximum of 6 hops. In reality, using our ERPL a node can have any number of neighbors, and similarly a node can have any number of hops along the forwarding path to the gateway. Each node in a network generates 1000 P2P communication packets, and for each packet a node randomly selects a peer node. Each simulation scenario was repeated 30 times. Fig. 4 (a) shows ERPL and RPL total number of P2P transmissions comparison. The figure shows that ERPL demonstrates statistically significantly lower number of P2P transmissions to deliver P2P data packets. Fig. 4 (b) shows the difference between the two protocols P2P transmissions. It is evident that in all the cases ERPL demonstrates at-least 100,000 lower number of transmissions compared to RPL. Hence, our enhanced P2P routing and forwarding algorithms can help to reduce energy consumption, and it may positively impact the performance of MP2P and P2MP communications as well due to lower number of transmissions in a network.

V. PERFORMANCE EVALUATION

In this section, we present a simulation and physical testbed-based protocols performance evaluation. We implemented ERPL in the Contiki operation system. Therefore, simulation-based performance evaluations were carried out using the widely used Cooja emulator [13] using the real programing code for embedded devices. The testbed based performance evaluation was performed using the FlockLAB [14] testbed. The protocols use the objective function 0 [12]. In our Cooja setup, 50 Tmote sky motes were placed in a network that spans an area of $250 \times 250 \ m^2$. There is a single gateway in our network. We evaluate the protocols performance using different random and grid topologies. Moreover, we analyse the impact of the gateway's position and the nodes' data generation interval on the protocols performance. The general simulation parameters are stated in Table I. The total duration of a single simulation is 1020 seconds. We categorize our performance benchmarks as follows: reliability, latency, and energy consumption. For reliability we measure and report mean packet delivery ratio (PDR), for latency we report mean per-packet end-to-end delay, and for energy consumption we report the total number of times P2P communication packets are relayed and the total number of retransmissions. We compare our ERPL protocol
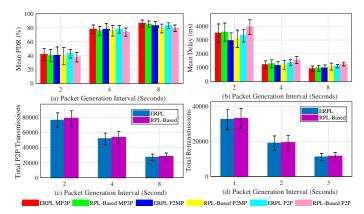

Fig. 5. Protocols' Performance with Different Random Topologies (Gateway at Network Edge)

TABLE I
COOJA SIMULATION PARAMETERS

Parameter	Value
MAC layer	IEEE 802.15.4 CSMA-CA
MAC layer ACKs	Enabled
Radio model	Unit disk with distance loss
Channel rate	250 kbps
MAC layer queue size	10 frames
Node transmission range	50 meters
Node carrier sensing range	100 meters

against a RPL protocol implementation. The experiments were performed using storing MoP.

For MP2P communication, each node generates data packets. Nodes start generating data packets at 10 simulation seconds, and stop the generation of data packets at 1000 seconds. For P2MP communication, the gateway starts generating data packets at 20 seconds, and stops the generation of P2MP data packets at 1000 seconds. For each data packet, the gateway selects a random destination node. Similarly, for P2P communication, each node starts P2P communication at 20 seconds, and stops P2P packet transmission at 1000 seconds. For each data packet, nodes randomly select the peer destination node. We repeated our experiments by varying the MP2P, P2MP, and P2P communications packet generation interval. The packet generation interval is varied from 2 seconds to 8 seconds.

A. Random Topology Results

To analyze the protocols performance we used 10 different fully connected random topologies. Moreover, to analyze the impact of the gateway position on the two protocols performance, we carried out two different set of experiments. In the first set of experiments, the gateway is placed at the edge of the networks and in the second set of experiments the gateway is placed at the center of the networks.

Fig. 5 shows the protocols performance in random topologies when the gateway is placed at the edge of the networks. Fig. 5 (a) shows the mean PDR achieved by the protocols in MP2P, P2MP, and P2P communications along with the 95% confidence interval (CI) around the mean based on 10 different random topologies. Regardless of the packet generation interval, our ERPL protocol achieves higher mean PDR compared to RPL-based protocol. In all random topologies,

ERPL achieved higher PDR corresponding to the different communication types, however the CIs for both protocols overlap. Due to different random topologies, the two protocols performance vary from topology to topology, hence the CIs overlap regardless of the fact that ERPL achieved better PDR in each random topology. In general, with an increase in the packet generation interval, the protocols demonstrate better PDR due to lower data traffic load in the networks. ERPL focuses on enhancing P2P routing and forwarding, therefore it outperforms standard RPL by a higher margin in P2P data forwarding compared to the other two types of communication. Fig. 5 (a) also shows that our enhanced P2P data forwarding mechanism also positively impacts MP2P and P2MP communications performance.

Fig. 5 (b) shows the mean per-packet end-to-end delay demonstrated by both protocols in MP2P, P2MP, and P2P communications along with the 95% CI for the mean based on 10 different random topologies. In general, ERPL demonstrates lower mean per-packet delay for the different types of communication compared to standard RPL. For the different types of communication, the CIs corresponding to both protocols overlap regardless of the fact that ERPL demonstrates lower delay in the different random topologies. This is again due to the fact that the protocols performance vary from topology to topology, hence the CIs overlap. With an increase in the data packet generation interval both protocols demonstrate lower delay because of relatively lower data traffic in the networks. As was the case with PDR, our improved P2P data packet forwarding mechanism also results in a lower per-packet end-to-end delay corresponding to MP2P and P2MP communications.

Fig. 5 (c) and Fig. 5 (d) show the protocols comparison with regard to the number of times P2P packets are relayed in reaching their destination node, and the total number of retransmissions respectively. The number of times P2P packets are relayed in network with ERPL is lower than for the standard RPL protocol. Moreover, ERPL also demonstrates fewer retransmissions. As ERPL relays P2P packets less times, it results in lower contention in the networks, hence ERPL achieves better PDR, delay, and total number of retransmissions.

Fig. 6 shows the two protocols performance in random topologies when the gateway is placed at the center of the network topology. Fig. 6 (a) shows the mean PDR achieved by the protocols in MP2P, P2MP, and P2P communications along with the 95% confidence interval (CI) for the mean based on 10 different random topologies. The general trends shown in Fig. 6 (a) are similar to the trends shown in Fig. 5 (a), i.e. ERPL demonstrates higher PDR in MP2P, P2MP, and P2P communications (again CIs overlap due to the protocols performance variation with each topology), and ERPL outperforms the standard RPL protocol in P2P communication by a higher margin compared to the other two communication types. However, comparison of Fig. 5 (a) and Fig. 6 (a) reveals an interesting insight, i.e. the PDR achieved by the protocols corresponding to MP2P and P2MP communications is slightly increased and mostly the PDR achieved by the protocols cor-
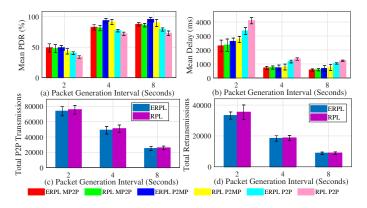

Fig. 6. Protocols' Performance with Different Random Topologies (Gateway at Network Center)

responding to P2P communication is slightly decreased when the gateway is at the center of the networks compared to the scenario when the gateway was at the edge of the networks. Placing the gateway at the center of the network on average results in shorter MP2P and P2MP data forwarding paths, hence better performance. However, shorter MP2P and P2MP forwarding paths imply that on average each node's DODAG has a lower number of nodes, hence many P2P data packets go through the gateway. This results in relatively longer P2P data forwarding paths and the longer paths negatively impact P2P communications performance. Similarly, placing the gateway at the edge of the network on average results in longer MP2P and P2MP data forwarding paths, hence MP2P and P2MP communication PDR is negatively impacted. However, longer paths result in relatively bigger DODAG at each node, hence a lower number of P2P packets go through the gateway, and it results in better P2P communication PDR.

Fig. 6 (b) shows the mean per-packet end-to-end delay achieved by the protocols in MP2P, P2MP, and P2P communications along with the 95% CI for the mean when the gateway was placed at the center of 10 different random network topologies. The general trends shown in Fig. 6 (b) are similar to the trends shown in Fig. 5 (b), and the only difference is that due to relatively shorter data forwarding paths MP2P and P2MP communication data packets experience lower delay.

Fig. 6 (c) and Fig. 6 (d) show the protocols comparison with regard to the number of times P2P packets are relayed to reach their destination node and the total number of retransmissions respectively. The trends shown in Fig. 6 (c) and 6 (d) are similar to the trends shown in Fig. 5 (c) and 5 (d) respectively. ERPL relays P2P packets fewer times, hence it demonstrates better PDR, delay, and total number of retransmissions.

B. Results Using Grid Network Topology

Here we present both protocols performance based on a grid network topology. Fig. 7 shows the two protocols performance based on a grid topology when the gateway is placed at the center of the network. Fig. 7 (a) shows the mean PDR demonstrated by the protocols in MP2P, P2MP, and P2P communications. Compared to standard RPL, our ERPL demonstrates higher PDR in all cases. The performance difference among the evaluated protocols increases as we decrease
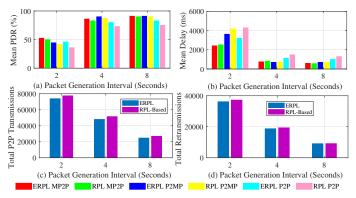
Fig. 7. Protocols' Performance with Grid Topology (Gateway at Network Edge)

the data packet generation interval. The ERPL P2P forwarding algorithm tries to lowers the number of transmissions required to deliver a P2P data packet to its destination, hence the performance difference between the protocols increases as we decrease the data packet generation interval. ERPL's enhanced P2P routing and forwarding algorithm also positively impacts MP2P and P"MP communication performance, which is similar to the random network topologies results.

Fig. 7 (b) shows the mean per-packet end-to-end delay achieved by the protocols with MP2P, P2MP, and P2P communications. In all cases, ERPL demonstrates lower delay compared to standard RPL. The performance gap between the protocols increases as we decrease the packet generation interval. The reason for this is the same as discussed for the PDR results. With an increase in the packet generation interval both protocols demonstrate lower delay due to lower data traffic load in the network.

Fig. 7 (c) and Fig. 7 (d) show the two protocols performance with regard to the number of times P2P packets are relayed in the network and the total number of retransmissions. The number of times P2P packets are relayed in the network using ERPL is lower compared to standard RPL. Moreover, ERPL demonstrates fewer retransmissions. As ERPL relays P2P packets less, this results in a lower contention, hence ERPL demonstrates better PDR, delay, and total number of retransmissions in the grid network topology as well.

Fig. 8 shows the two protocols performance based on the grid network topology when the gateway is placed at the network's center. Fig. 8 (a) shows the two protocols' PDR. The trends shown in Fig. 8 (a) are similar to the trends shown in Fig. 7 (a). Comparison of Fig. 7 (a) and Fig. 8 (a) reveals that placing the gateway at the center of the network slightly improves MP2P and P2MP communications' PDR, however the PDR corresponding to P2P communication is negatively impacted. This happens exactly due to the same reason that we discussed above for random network topologies.

Fig. 8 (b) shows the mean per-packet end-to-end delay achieved by the protocols with MP2P, P2MP, and P2P communications. The trends shown in Fig. 8 (b) are similar to the trends shown in Fig. 7 (b). However, MP2P and P2MP communications experience lower delay compared to the delay experienced by the communications when the gateway

was at the edge of the network. This is due to the fact that on average MP2P and P2MP forwarding paths are shorter when the gateway is at the center of the network.

Fig. 8 (c) and Fig. 8 (d) show the protocols comparison with regard to the number of times P2P packets are relayed in the network and the total number of retransmissions. The trends shown in Fig. 8 (c) and 8 (d) are similar to the trends shown in Fig. 7 (c) and Fig. 7 (d) respectively. Fewer total P2P transmissions associated with ERPL results in the protocol's superior performance overall.

C. Testbed Results

To validate our simulation-based results, we evaluated the protocols using the FlockLAB [14] testbed. The experiments were performed using the FlockLAB's TelosB motes. In the testbed-based experiments, each experiment duration and traffic generation model were the same as we used in our simulation based experiments. The current FlockLAB deployment consists of 30 nodes, however at the time of our experiments only twenty seven TelosB motes were operational. Twenty three nodes were deployed indoors across one floor in an office building, distributed in offices, hallways and store rooms. Four nodes were deployed outside, placed on the roof of an adjacent building a few meters beneath the floor with the indoor nodes. In the testbed based experiments, a node at the edge of the network was selected as the gateway.

Fig. 9 (a) shows the PDR achieved by the protocols. The PDR results obtained through the testbed validate our simulation based PDR results as the trends are similar, i.e. in all cases ERPL outperforms the standard RPL protocol. With a decrease in the packet generation interval the performance gap between the two protocols increases and both protocols show higher PDR with an increase in the packet generation interval.

Fig. 9 (b) shows the delay demonstrated by the two protocols. The delay results obtained through the testbed again validate the results obtained through simulations as the trends are similar, i.e. in all cases ERPL outperforms the standard RPL protocol. Due to lower contention with an increase in the packet generation interval both protocols demonstrate lower delay.

Fig. 9 (c) and Fig. 9 (d) show the two protocols performance with regard to the number of times P2P packets are relayed in the network and the total number of retransmissions. The testbed results validate the simulation based results as again the trends are similar, i.e. ERPL demonstrates lower total P2P transmissions and retransmission. Lower P2P transmissions associated with ERPL result in the protocol's improved performance in all three communication types.

D. Discussion

Our ERPL reduce the number of times a P2P packet is relayed in a network to reach its destination. This enhancement is achieved without any additional control overhead, hence ERPL reduces the amount of contention in a network. This not only positively impacts the performance of P2P communication, but also positively impacts the performance
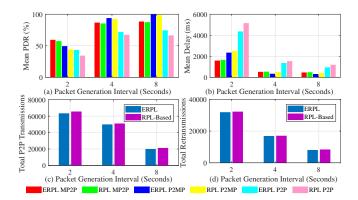
Fig. 8. Protocols' Performance with Grid Topology (Gateway at Network Center)
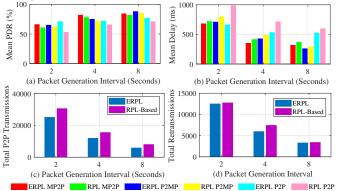

Fig. 9. Protocols' Performance in FlockLAB Testbed

of MP2P and P2MP communications. The protocol's better P2P routing and forwarding algorithms result in lower energy consumption as the protocol also demonstrates fewer retransmissions. Placing the gateway at the edge of the network results in longer MP2P and P2MP data forwarding paths, hence MP2P and P2MP performance is negatively impacted. However, at the same time it results in a relatively bigger DODAG at each node, therefore nodes can discover more peer nodes. Consequently, the number of times a P2P packet is relayed through the gateway is lowered. Similarly, placing the gateway at the center of a network results in shorter MP2P and P2MP data forwarding paths. It improves MP2P and P2MP communications, however due to a relatively shallow DODAGs most P2P packets travel through the gateway, hence P2P communication is negatively impacted.

VI. Conclusions

Standard RPL exhibits poor performance in P2P communication due to the required additional control overhead and inefficient P2P forwarding. Here, we presented ERPL, an enhanced routing and forwarding mechanism for P2P communication for RPL's storing and non-storing MoPs. ERPL reduces the number of transmissions required to delivery a P2P data packet to the destination node thus reducing delay and supporting delay sensitive Cyber-Physical Systems applications over LLNs. Moreover, ERPL can coexist with existing RPL implementations. Our results demonstrated that ERPL not only outperforms the standard RPL protocol in P2P communications, but it also outperforms the protocol in MP2P and P2MP communications.

References

[1] 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," March 2012, RFC 6550.

[2] J. Ko, J. Jeong, J. Park, J. A. Jun, O. Gnawali, and J. Paek, "DualMOP-RPL: Supporting Multiple Modes of Downward Routing in a Single RPL Network," *ACM Transations on Sensor Networks*, vol. 11, no. 2, pp. 39:1–39:20, March 2015.

[3] W. Gan, Z. Shi, C. Zhang, L. Sun, and D. Ionescu, "MERPL: A More Memory-Efficient Storing Mode in RPL," in 19th *IEEE International Conference on Networks*, 2013, pp. 1–5.

[4] C. Kiraly, T. Istomin, O. Iova, and G. P. Picco, "D-RPL: Overcoming Memory Limitations in RPL Point-to-Multipoint Routing," in *IEEE 40th Conference on Local Computer Networks*, 2015, pp. 157–160.

[5] O. Iova, P. Picco, T. Istomin, and C. Kiraly, "RPL: The Routing Standard for the Internet of Things... Or Is It?" *IEEE Communications Magazine*, vol. 54, no. 12, pp. 16–22, 2016.

[6] K. Heurtefeux and 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.

[7] J. Tripathi, J. C. de Oliveira, and 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.

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

[9] M. O. Farooq, C. J. Sreenan, K. Brown, and T. Kunz, "RPL-Based Routing Protocols For Multi-Sink Wireless Sensor Networks," in 11th *IEEE International Conference on Wireless and Mobile Computing, Networking and Communications*, 2015, pp. 452–459.

[10] E. Baccelli, M. Philipp, and M. Goyal, "The P2P-RPL Routing Protocol for IPv6 Sensor Networks: Testbed experiments," in 19th *International Conference on Software, Telecommunication, and Computer Networks*, 2011, pp. 200–209.

[11] M. Goyal, E. Baccelli, M. Philipp, A. Brandt, and J. Martocci, "Reactive Discovery of Point-to-Point Routes in Low-Power and Lossy Networks," August 2013, RFC 6997.

[12] P. Thubert, "Objective Function Zero for the Routing Protocol for Low-Power and Lossy Networks (RPL)," March 2012, RFC 6552.

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

[14] R. Lim, F. Ferrari, M. Zimmerling, C. Walser, P. Sommer, and J. Beutel, "FlockLab: A Testbed for Distributed, Synchronized Tracing and Profiling of Wireless Embedded Systems ," in 12th *ACM/IEEE Conference on Information Procecssing in Wireless Sensor Networks (IPSN)*, 2013, pp. 153–165.