Networking Working Group
Internet-Draft
Intended status: Informational
Expires: February 17, 2012

J. Tripathi, Ed.
J. de Oliveira, Ed.
Drexel University
JP. Vasseur, Ed.
Cisco Systems, Inc.

August 16, 2011

Performance Evaluation of Routing Protocol for Low Power and Lossy Networks (RPL)
draft-tripathi-roll-rpl-simulation-07

Abstract

This document presents a performance evaluation of the Routing Protocol for Low power and Lossy Networks (RPL) for a small outdoor deployment of sensor nodes and for a large scale smart meter network. Detailed simulations are carried out to produce several routing performance metrics using these real-life deployment scenarios.

Status of this Memo

Copyright Notice

Contents

1 Terminology

Please refer to the following document for terminology in [I-D.ietf-roll-terminology]. In addition, the following terms are specified:

- PDR : Packet Delivery Ratio.
- CDF : Cumulative Distribution Function.
- Fractional Stretch Factor of link ETX Metric against ideal shortest path: The ETX path stretch is defined as the difference between the number of expected transmissions (ETX Metric) taken by a packet traveling from source to destination, following a route determined by RPL and a route determined by a hypothetical ideal shortest path routing protocol (using link ETX as the metric). The fractional path stretch is the ratio of ETX path stretch to ETX path cost for the shortest path route for that source- destination pair.
- Stretch factor for node hop distance against ideal shortest path: The hop stretch is defined as the difference between the number of hop counts taken by a packet traveling from source to destination, following a route determined by RPL and by a hypothetical ideal shortest path algorithm, both using ETX as the link cost. The fractional stretch factor is computed as the ratio of path stretch to count value between a source-destination pair for the hypothetical shortest path route optimizing ETX path cost.

2 Introduction

Designing a routing protocol for Low power and Lossy link Networks (LLNs) imposes great challenges, mainly due to low data rates, high probability of packet delivery failure, and strict energy constraint in nodes. The IETF ROLL Working Group took on this task and specified the Routing Protocol for Low power and Lossy Networks (RPL) in [I-D.ietf-roll-rpl].

RRPL is designed to meet the core requirements specified in [RFC5826],[RFC5867],[RFC5673] and [RFC5548].

This document's contribution is to provide a performance evaluation of RPL with respect to several metrics of interest. This is accomplished using real data and topologies in a discrete event simulator, developed to reproduce the protocol behavior.

The following metrics are evaluated:

- Path quality metrics;
- Control plane overhead;
- End to End delay between nodes.
- Ability to cope with unstable situations (link churns, node dying);
- Required resource constraints on nodes (routing table size, etc.).

Some of these metrics are mentioned in the aforementioned RFCs, whereas others have been introduced considering the challenges and unique requirements of LLNs, as mentioned in [I-D.ietf-roll-rpl]. For example, routing in a home automation deployment has strict time bounds on protocol convergence after any change in topology as mentioned in section 3.4 of [RFC5826]. [RFC5673] requires bounded and guaranteed end-to-end delay for routing in an industrial deployment and [RFC5548] requires comparatively loose bound on latency for end-to-end communication. [RFC5548] mandates scalability in terms of protocol performance for a network of size ranging from 10^2 to 10^4 nodes.

Although simulation cannot prove formally that a protocol operates properly in all situations, it can give a good level of confidence in protocol behavior in highly stressful conditions, if and only if real life data are used. Simulation is particularly useful when theoretical model assumptions may not be applicable to such networks and scenarios. In this document, real deployed network data traces have been used to model link behaviors and network topologies.

3 Methodology and Simulation Setup

In the context of this document, RPL has been simulated using OMNET++ [OMNETpp], a well-known discrete event based simulator written in C++ and NED. Castalia-2.2 [Castalia-2.2] has been used as Wireless Sensor Network Simulator framework within OMNET++. The output and events in the simulating are visualized with the help of the Network AniMator or NAM, which is distributed with NS (Network Simulator) [NS-2].

Note that NS or any of its versions are not used in this simulation study. Only the visualization tool was borrowed for verification purposes.

In contrast with theoretical models, which as stated before may have assumptions not applicable to lossy links, real-life data has been used for two aspects of the simulations:

- **Link failure model:** Time varying real network traces containing packet delivery probability for each link, over all channels for both indoor network deployment and outdoor network deployment.
- **Topology:** Gathered from real-life deployment (traces mentioned above) as opposed to random topology simulations.

A 45 node topology, deployed as an outdoor network, shown in Figure 1 and a 2442 node topology, gathered from a smart meter network deployment, were used in the simulations.

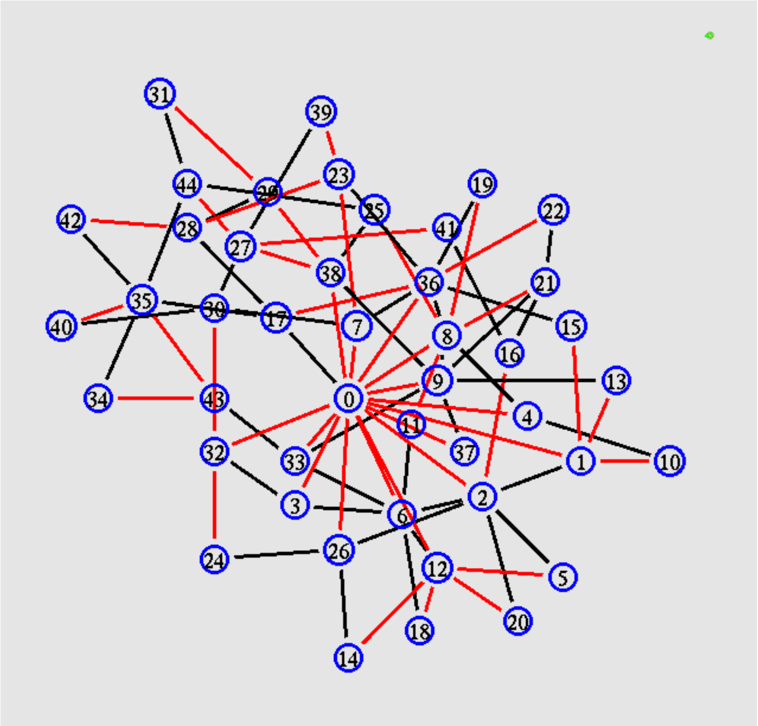

Figure 1: 45 nodes outdoor network topology.

Note that this is just a start to validate the simulation before using large scale networks.

A set of time varying link quality data was gathered from real network deployment to form a database used for the simulations. Each link in the topology randomly 'picks up' a link model (trace) from the database. Each link has a Packet Delivery Ratio (PDR) that varies with time (in the simulation, a new PDR is read from the database every 10 minutes) according to the gathered data. Packets are dropped randomly from that link with probability (1 - PDR). Each time a packet is about to be sent, the module generates a random number using the Mersenne Twister Random number generation method. The random number is compared to the PDR to determine whether the packet should be dropped. Note that each link uses a different random number generator to maintain true randomness in the simulator, and to avoid correlation between links. Also, the packet drop applies to all kinds of data and control packets (RPL) such as the DIO, DAO, DIS packets defined in [I-D.ietf-roll-rpl]. Figure 2 shows a typical temporal characteristic of links from the indoor network traces used in the simulations. The figure shows several links with perfect connectivity, some links with PDR as low as 10% and several for which the PDR may vary from 30% to 80%, shaprply changing back and forth between high value (strong connectivity) and low value (weak connectivity).

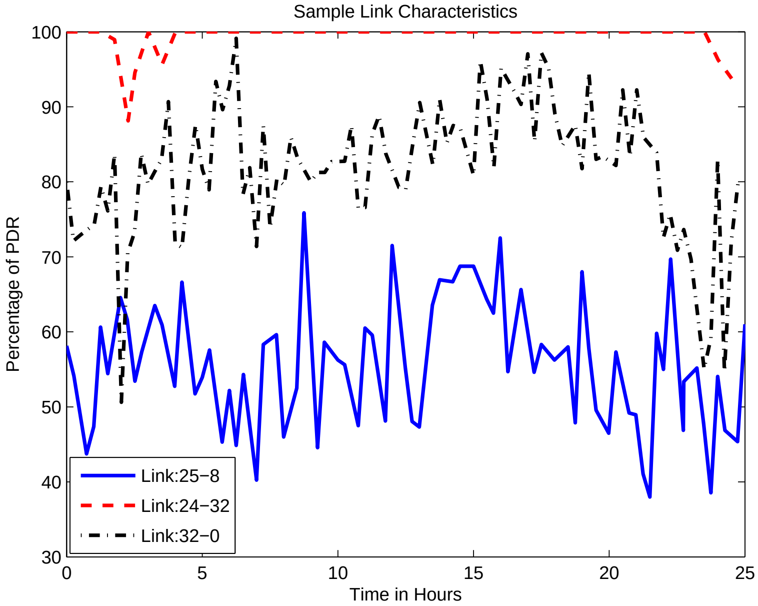

Figure 2: Example of link characteristics.

In the RPL simulator, the LBR first initiates sending out DIO messages, and the DAG is gradually constructed. RPL makes use of trickle timers: the protocol sets a minimum time period, with which the nodes start re-issuing DAOs, and this minimum period is denoted by the parameter I_{min}. RPL also sets an upper limit on how many times this time period can be doubled, and is denoted by the parameter $I_{doubling}$, as defined in [I-D.ietf-roll-rpl]. For the simulation, I_{min} is initially set to 1 second and $I_{doubling}$ is equal to 16, so that maximum time between two consecutive DIO emissions by a node (under a steady network condition) is 18.2 hours. The trickle time interval for emitting DIO message assumes the initial value of 1 second, and then changes over simulation time as mentioned in [I-D.ietf-roll-trickle].

Another objective of this study is to give insight to the network administrator on how to tweak the trickle values. These recommendations could then be used in applicability statement documents.

Each node in the network, other than the LBR (Low power and lossy Border Router), also emits DAO messages as specified in [I-D.ietf-roll-rpl], to initially populate the routing tables with the prefixes received from children via the DAO messages to support Point to Point (P2P) and Point to Multipoint traffic (P2MP) in the "down" direction. During these simulations, it is assumed that each node is capable of storing route information for other nodes in the network (storing mode of RPL).

For nodes implementing RPL, as expected, the routing table memory requirement varies according to the position in the DODAG (Destination Oriented Directed Acyclic Graph). The (worst-case) assumption is made that there is no route summarization (aggregation) in the network. Thus a node closer to the DAG will have to store more entries in its routing table. It is also assumed that all nodes have equal memory capacity to store the routing states.

For simulations of the indoor network, each node sends traffic according to a Constant Bit Rate (CBR) to all other nodes in the network, over the simulation period. Each node generates a new data packet every 10 seconds. Each data packet has a size of 127 bytes including 802.15.4 PHY/MAC headers and RPL packet headers. All control packets are also encapsulated with 802.15.4 PHY/MAC headers. To simulate a more realistic scenario, 80% of the generated packets by each node are destined to the root, and the remaining 20% of the packets are uniformly assigned as destined to nodes other than the root. Therefore the root receives a considerably larger amount of data than other nodes. These values may be revised when studying P2P traffic so as to have a majority of traffic going to all nodes as opposed to the root. In the later part of the simulation, a typical home/building routing scenario is also simulated and different path quality metrics are computed for that traffic pattern.

The packets are routed through the DODAG built by RPL according to the mechanisms specified in [I-D.ietf-roll-rpl].

A number of RPL parameters are studied (such as Packet Rate from each source, Time Period of the LBR emitting new DAG Sequence Number) to observe their effect on the performance metric of interest.

4 Performance Metrics

4.1 Common Assumptions

As the DAO messages are used to feed the routing tables in the network, they grow with time and size of the network. Nevertheless, no constraint was imposed on the size of the routing table nor on how much information the node can store. The routing table size is not expressed in terms of Kbyte of memory usage but measured in terms of number of entries for each node. Each entry has the next hop node and path cost associated with the destination node.

The link ETX (Expected Transmission Count) metric is used to build the DODAG. The link ETX routing metric is specified in [I-D.ietf-roll-routing-metrics].

4.2 Path Quality

Number of Hops: For each source-destination pair, the average number of hops for both RPL and shortest path routing is computed. Shortest path routing refers to a hypothetical ideal routing protocol that would always provide the shortest path in term of path cost ETX (or whichever metric is used) in the network. The Cumulative Distribution Function (CDF) of hop distance for all paths ($n{*}(n-1)$ in an n-node network) in the network with respect to the number of hops is plotted in Figure 3 for both RPL and shortest path routing. One can observe that the CDF corresponding to 4 hops is around 80% for RPL and 90% for shortest path routing. In other words, for the given topology, 90% of paths have a path length of 4 hops or less with an ideal shortest path routing methodology, whereas in RPL Point-to- Point (P2P) routing, 90% of the paths will have a length of no more than 5 hops. This result indicates that despite having a non-optimized P2P routing scheme, the path quality of RPL is close to an optimized P2P routing mechanism for the topology in consideration. Another reason for this may relate to the fact that the sink is at the center of the network, thus routing through the sink is often close to an optimal (shortest path) routing. This result may be different in a topology where the sink is located at one end of the network.

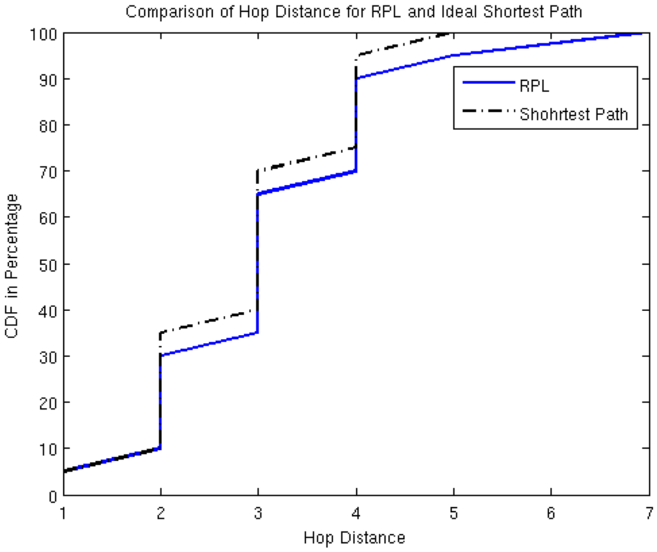

Figure 3: CDF: hop distance versus number of hops.

Path Cost ETX: The path cost ETX along the path is computed for each source-destination pair. In the simulation, the path ETX from source to destination for each packet is computed. Figure 4 shows the CDF of the total path cost ETX, both with RPL and shortest path routing. Here also one can observe that the path cost ETX from all source to all destinations is close to that of a shortest path routing for the network.

Path Stretch: In this simulation, the path stretch is also calculated for each packet that traversed the network. The path stretch is determined as the difference between the number of hops taken by a packet while following a route built via RPL and the number of hops taken by shortest path routing (using link ETX as the metric).

Once again, the CDF of the path stretch is plotted against the value of path stretch over all packets in Figures 5 and 6, for hop count stretch and ETX metric stretch, respectively. It can be observed that for a few packets, the path built via RPL has fewer hops than the ideal shortest path where path ETX is minimized along the DAG. This is because there are a few source-destination pairs where the total path ETX is equal to or less than that of the ideal shortest path when the packet takes a longer hop count. As the RPL implementation ignores 20% change in total path cost before switching to a new parent or emitting new DIO, RPL not necessarily provides the shortest path in terms of total ETX path cost. Thus, this implementation yields a few paths with smaller hop count but larger (or equal) total ETX path cost.

The data for the CDF of hop count and path cost ETX for the ideal shortest path (SP) and a path built via RPL, along with the CDF of the routing table size is given below in Table 1. Figures 3 to 7 relate to the data in this table.

Overall, the path quality metrics give us important information about the protocol's performance when minimizing the path cost ETX is the objective to form the DAG. The protocol, as explained, does not

CDF (%age)	Hop (SP)	Hop (RPL)	ETX Cost (SP)	ETX Cost (RPL)	Routing Table Size
0	1.0	1.0	1	1.0	0
5	1.0	1.03	1	1.242	1
10	2.0	2.0	2	2.048	2
15	2.0	2.01	2	2.171	2
20	2.0	2.06	2	2.400	2
25	2.0	2.11	2	2.662	3
30	2.0	2.42	2	2.925	3
35	2.0	2.90	3	3.082	3
40	3.0	3.06	3	3.194	4
45	3.0	3.1	3	3.41	4
50	3.0	3.15	3	3.626	4
55	3.0	3.31	3	3.823	5
60	3.0	3.50	3	4.032	6
65	3.0	3.66	3	4.208	7
70	3.0	3.92	4	4.474	7
75	4.0	4.16	4	4.694	7
80	4.0	4.55	4	4.868	8
85	4.0	4.70	4	5.091	9
90	4.0	4.89	4	5.488	10
95	4.0	5.65	5	5.923	12
100	5.0	7.19	9	10.125	44

Table 1: Path Quality CDFs

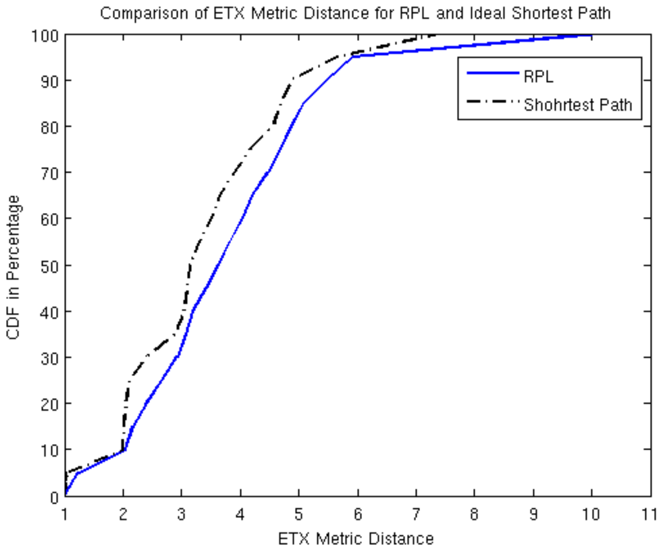

Figure 4: CDF: Total ETX along path versus ETX value.

always provide an optimum path, especially for peer-to-peer communication. However, it does end-up reducing the control overhead cost, reducing unnecessary parent selection and DIO message forwarding events, by choosing a non-optimized path. Despite this specific implementation technique, around 30% of the packets travel the same number of hops as an ideal shortest path routing mechanism, and 20% of packets experience the same number of attempted transmissions to reach the destination. On average, this implementation costs only a few extra transmission attempts and saves a large number of control packet transmissions.

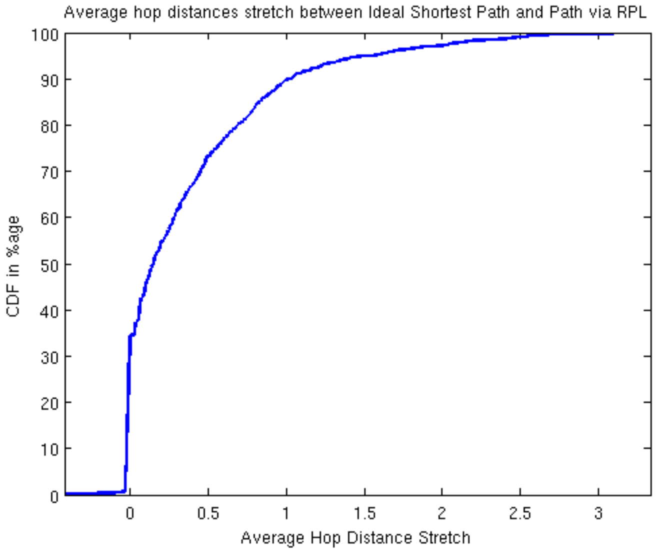

Figure 5: CDF: Hop count stretch versus hop count of a packet.

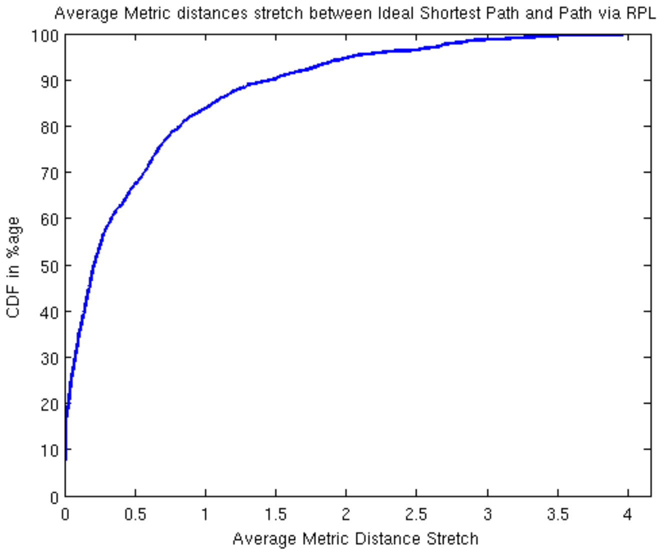

Figure 6: CDF: ETX metric stretch versus ETX value.

4.3 Routing Table Size

The objective of this metric is to observe the distribution of the number of entries per node. Figure 7 shows the CDF of the number of routing table entries for all nodes. Note that 90% of the nodes need to store less than 10 entries in their routing table. This result shows that the protocol is capable of accommodating devices with low storage capacity. This feature has been mandated in RFC 5826[RFC5826]and RFC 5867 [RFC5867].

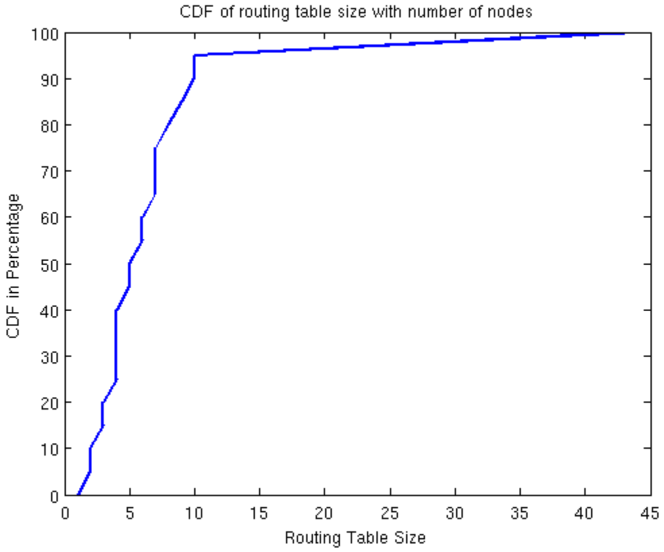

Figure 7: CDF of routing table size with respect to number of nodes.

4.4 Delay bound for P2P Routing

For delay sensitive applications, such as home and building automation, it is critical to optimize the end-to-end delay. Figure 8 shows the upper bound and distributions of delay for paths between any two given nodes for different hop counts between source and destination. Here, the hop count refers to the number of hops a packet travels to reach the destination when using RPL paths. This hop distance does not correspond to shortest path distance between two nodes. Note that, each packet has a length of 127 bytes, with a 240 kbps radio, which makes the transmission time approximately 4 ms.

RFCs 5673[RFC5673] and 5548[RFC5548]mention a requirement for the end-to-end delivery delay to remain within a bounded latency. For instance, according to the industrial routing requirement, non-critical closed-loop applications may have a latency requirement that can be as low as 100 milliseconds, whereas monitoring services may tolerate a delay in the order of seconds. The results show that about 99% of the end-to end communication (where maximum hop-count is 7 hops) are bounded within the 100 ms requirement.

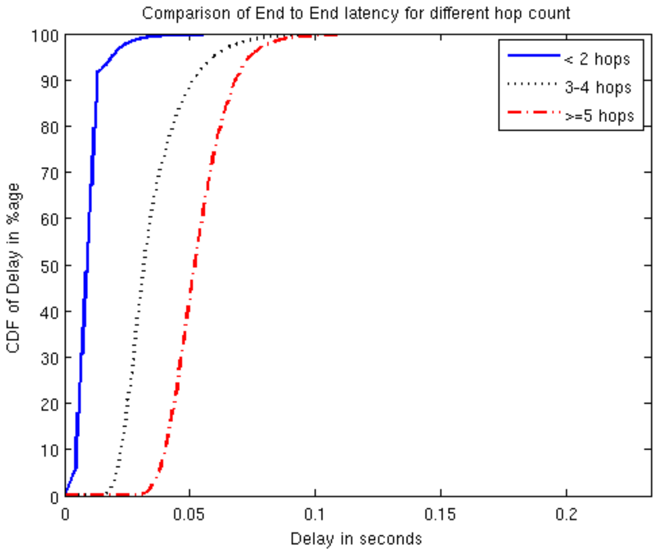

Figure 8: Comparison of packet latency for different path length expressed in hop count.

4.5 Control Packet Overhead

The control plane overhead is an important routing characteristic in LLNs. It is imperative to bound the control plane overhead. One of the distinctive characteristics of RPL is that it makes use of trickle timers so as to reduce the number of control plane packets by eliminating redundant messages. The aim of this performance metric is thus to analyse the control plane overhead both in stable conditions (no network element failure overhead) and in the presence of failures.

Data and control plane traffic comparison for each node: Figure 9 shows the comparison between the amount of data packets transmitted (including forwarded) and control packets (DIO and DAO messages) transmitted for all individual nodes when link ETX is used to optimize the DAG. As mentioned earlier, each node generates a new data packet every 10 seconds. Here one can observe that a considerable amount of traffic is routed through the sink itself. The x axis indicates the node ID in the network. Also, as expected, the nodes closer to sink and that act as routers (as opposed to leaves) handle much more data traffic than other nodes. Looking at a link close to the root, we can observe that the proportion of control traffic is negligible. This result also reinforces the fact that the amount of control plane traffic generated by RPL is negligible. Leaf nodes have comparable amount of data and control packet transmission (they do not take part in routing the data).

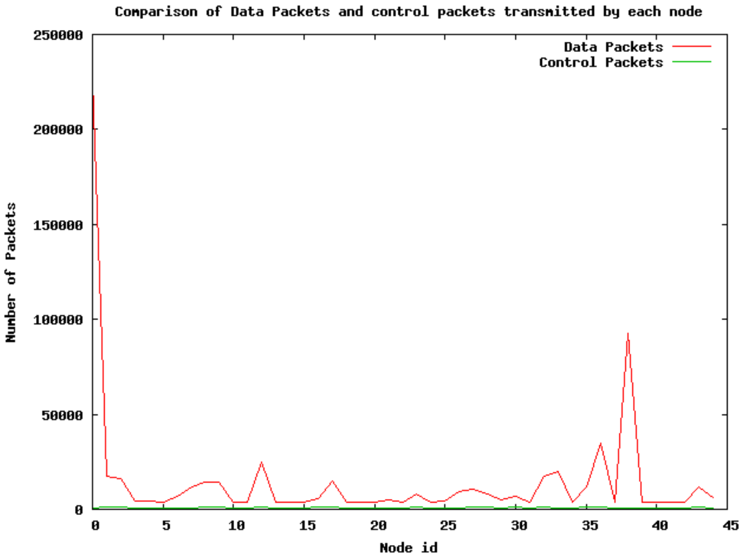

Figure 9: Amount of data and control packets transmitted against node ID using ETX as routing metric.

Data and Control Packet Transmission with Respect to Time: In Figures 10, 11 and 12, the amount of data and control packets transmitted for node 12 (low rank in DAG, closer to the root), node 43 (in the middle) and node 31 (leaf node) are shown, respectively. These values stand for number of data and control packets transmitted for each 10 minute intervals for the particular node, to help understand what is the ratio between data and control packets exchanged in the network. One can observe that nodes closer to the sink have a higher proportion of data packets (as expected), and the proportion of control traffic is negligible in comparison with the data traffic. Also, the amount of data traffic handled by a node within given interval varies largely over time for a node closer to sink, because in each interval the destinations of the packets from same source changes, while 20% of the packets are destined to the sink. As a result, pattern of the traffic handled changes widely in each interval for the nodes closer to the sink. For the nodes that are farther away from sink, the ratio of data and control traffic is smaller since the amount of data traffic is greatly reduced.

The control traffic load exhibits a wave-like pattern. The amount of control packets for each node drops quickly as the DODAG stabilizes due to the effect of trickle timers. However, when a new DODAG sequence is advertised (global repair of the DODAG), the trickle timers are reset and the nodes start emitting DIOs frequently again to rebuild the DODAG. For a node closer to the sink, the amount of data packets is much larger than that of control packets, and somewhat oscillatory around a mean value. The amount of control packets exhibits a ‘saw-tooth’ behavior. As the ETX link metric was used, when the PDR changes, the ETX link metric for a node to its child changes, which may lead to choosing a new parent, and changing the DAG rank of the child. This event resets the trickle timer and triggers the emission of a new DIO. Also, issue of a new DODAG sequence number triggers DODAG re-computation and resets the trickle timers. Therefore, one can observe that the number of control packets attains a high value for one interval, and comes down to lower values for subsequent intervals. The interval with high value of control packets denote the interval where the timers to emit new DIO are reset more frequently. As the network stabilizes, the control packets are less dense in volume. For leaf nodes, the amount of control packets is comparable to that of data packets, as leaf nodes are more prone to face changes in their DODAG rank as opposed to nodes closer to sink when the link ETX value in the topology changes dynamically.

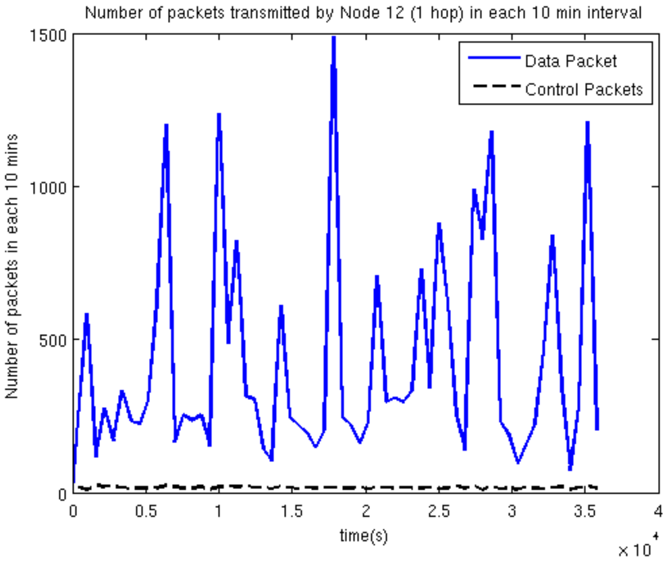

Figure 10: Amount of data and control packets transmitted for node 12.

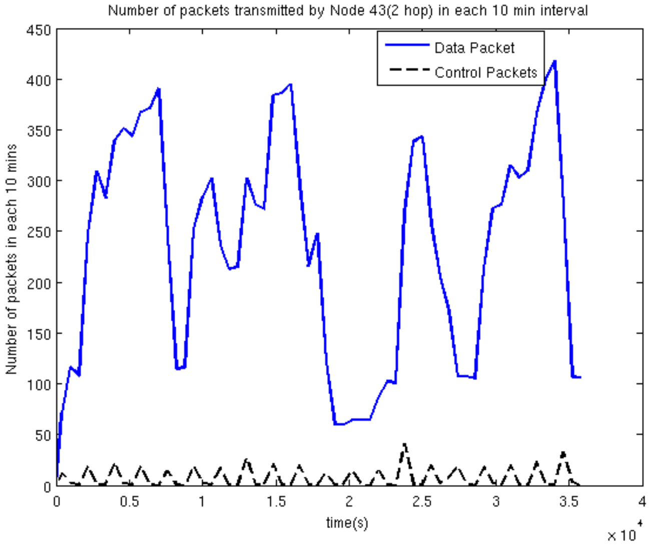

Figure 11: Amount of data and control packets transmitted for node 43.

4.6 Loss of connectivity

Upon link failures, a node may lose his parents: preferred and backup (if any) thus to leading to a loss of connectivity (no path to the DODAG root). RPL specifies two mechanisms for DODAG repairs, referred to as the global repair and local repair. In this version of the document, simulation results are presented to evaluate the amount of time data packets are dropped due to a loss of connectivity for the following

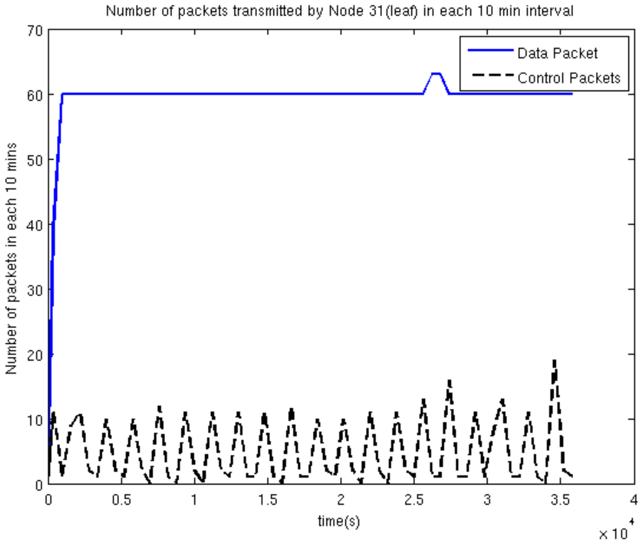

Figure 12: Amount of data and control packets transmitted for node 31.

two cases: *a*) wwhen only using global repair (i.e., the DODAG is rebuilt thanks to the emission of new DODAG sequence numbers by the DODAG root), and *b*) when using local repair (poisoning the sub-DAG in case of loss of connectivity) in addition to global repair. The idea is to tune the frequency at which new DODAG sequence numbers are generated by the DODAG root, and also to observe the effect of varying the frequency for global repair and the concurrent use of global and local repair. It is expected that more frequent increments of DODAG sequence number will lead to shorter duration of connectivity loss at a price of a higher rate of control packet in the network. For the use of both global and local repair, the simulation results show the trade-off in amount of time that a node may remain without service and total number of control packets for extra bit of signalling.

Figure 13 shows the CDF of time spent by any node without service, when the rate of data packet is one packet every 10 seconds, and new DODAG Sequence Number is generated every 10 minutes. This plot reflects the property of global repair without any local repair scheme. When all the parents are temporarily unreachable from a node, the time before it hears a DIO from another node is recorded, which gives the time without service. We define DAG repair timer to be the interval at which the LBR increments the DAG sequence number, thus triggering a global reoptimization. In some cases, this value might go up to the DAG repair timer value, because until a DIO is heard, the node does not have a parent, and hence no route to the LBR or other nodes not in its own sub-DAG. Clearly, this situation indicates a lack of connectivity and loss of service for the node.

The effect of the DAG repair timer on time without any service is plotted in Figure 14, where the source rate is 20 seconds/packet and in Figure 15, where the source sends a packet every 10 seconds.

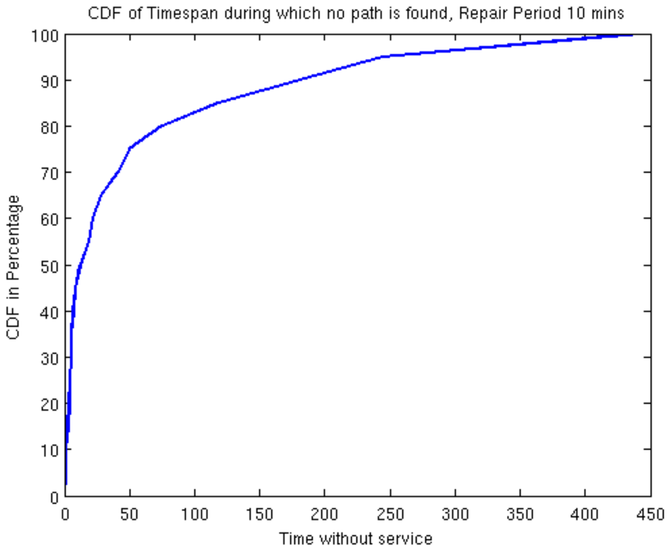

Figure 13: CDF: Loss of connectivity with global repair.

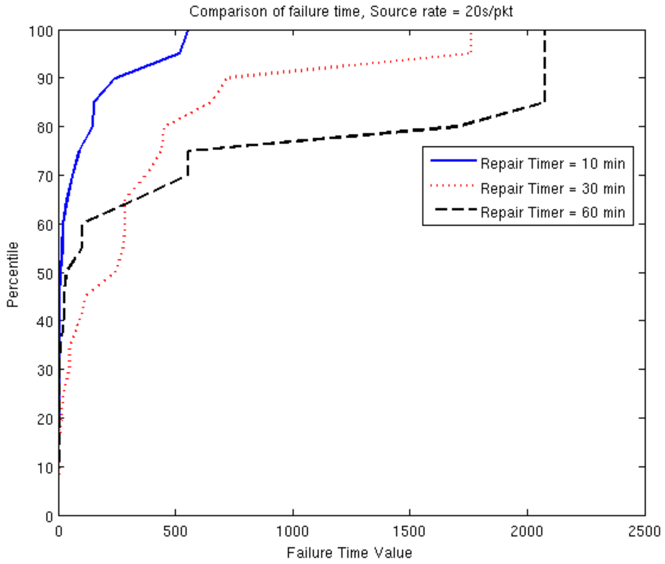

Figure 14: CDF: Loss of connectivity for different global repair period, packet rate 20/s.

The data for Figures 13 and 15 can be found below in the Table 2. The table shows how the CDF of time without connectivity to LBR increases while we increase the time period to emit new DAG sequence number, when the nodes generate a packet every 10 seconds.

The data for Figure 14 can be found in the Table 3. The table shows how the CDF of time without connectivity to LBR increases while we increase the time period to emit new DAG sequence number, when the nodes generate a packet every 20 seconds.

Figure 16 shows the effect of DAG global repair timer period on control traffic. As expected, as the frequency at which new DAG sequence numbers are generated increases, the amount of control traffic decreases because DIO messages are sent less frequently to rebuild the DODAG. However reducing the

CDF (%age)	Repair Period 10 Minutes	Repair Period 30 Minutes	Repair Period 60 Minutes
0	0.464	0.045	0.027
5	0.609	0.424	0.396
10	1.040	1.451	0.396
15	1.406	3.035	0.714
20	1.934	3.521	0.714
25	2.113	5.461	1.856
30	3.152	5.555	1.856
35	3.363	7.756	6.173
40	4.9078	8.604	6.173
45	8.575	9.181	14.751
50	9.788	21.974	14.751
55	13.230	30.017	14.751
60	17.681	31.749	16.166
65	29.356	68.709	16.166
70	34.019	92.974	302.459
75	49.444	117.869	302.459
80	75.737	133.653	488.602
85	150.089	167.828	488.602
90	180.505	271.884	488.602
95	242.247	464.047	488.602
100	273.808	464.047	488.602

Table 2: Loss of Connectivity time, Data rate - 1 Packet / 10 Seconds

CDF (%age)	Repair Period 10 Minutes	Repair Period 30 Minutes	Repair Period 60 Minutes
0	0.071	0.955	0.167
5	0.126	2.280	1.377
10	0.403	2.926	1.409
15	0.902	3.269	1.409
20	1.281	16.623	3.054
25	2.322	21.438	5.175
30	2.860	48.479	5.175
35	3.316	49.495	10.30
40	3.420	93.700	25.406
45	6.363	117.594	25.406
50	11.500	243.429	34.379
55	19.703	277.039	102.141
60	22.216	284.660	102.141
65	39.211	285.101	328.293
70	63.197	376.549	556.296
75	88.986	443.450	556.296
80	147.509	452.883	1701.52
85	154.26	653.420	2076.41
90	244.241	720.032	2076.41
95	518.835	1760.47	2076.41
100	555.57	1760.47	2076.41

Table 3: Loss of Connectivity time, Data rate - 1 Packet / 20 Seconds

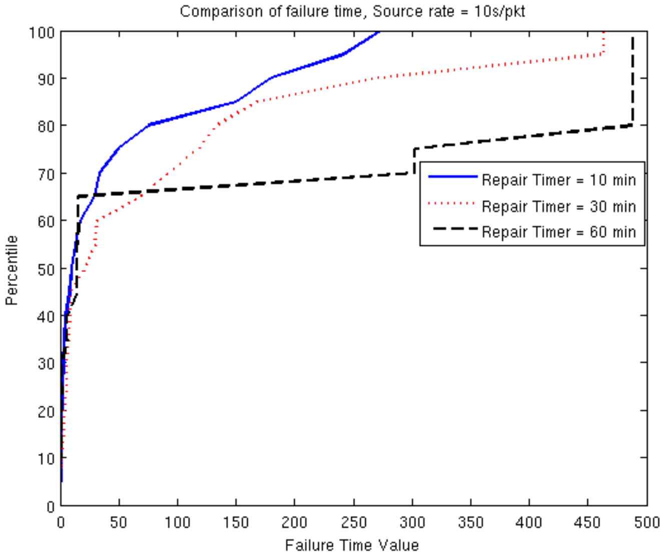

Figure 15: CDF: Loss of connectivity for different global repair period, packet rate 10/s.

control traffic comes at a price of increased loss of connectivity when only global repair is used.

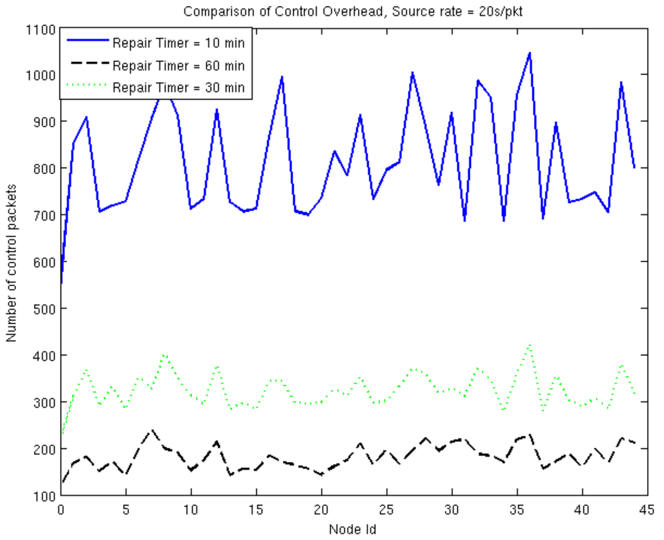

Figure 16: Amount of control traffic for different global repair periods.

From the above results, it is clear that the time the protocol takes to re-stablish routes and to converge, after an unexpected link or device failure happens, is fairly long. [RFC5826] mandates that “the routing protocol MUST converge within 0.5 seconds if no nodes have moved”. Clearly, implementation of a repair mechanism based on new DAG sequence number alone would not meet the requirements. Hence a local repair mechanism, in form of poisoning the sub-DAG and issuing DIS, has been adopted.

The effect of the DAG Repair Timer on time without service when local repair is activated is now observed and plotted in Figure 17, where the source rate is 20 seconds/packet. A comparison of the CDF of loss of connectivity for global repair mechanism and global + local repair mechanism is shown in Figures 18 and 19 (semilog plots, x axis in logarithmic and y axis in linear scale), where the source generates a packet every 10 seconds and 20 seconds, respectively. For these plots, the x axis shows time in log scale, and y axis denotes the corresponding CDF in linear scale. One can observe that using local repair (with poisoning of the sub-DAG) greatly reduces loss of connectivity.

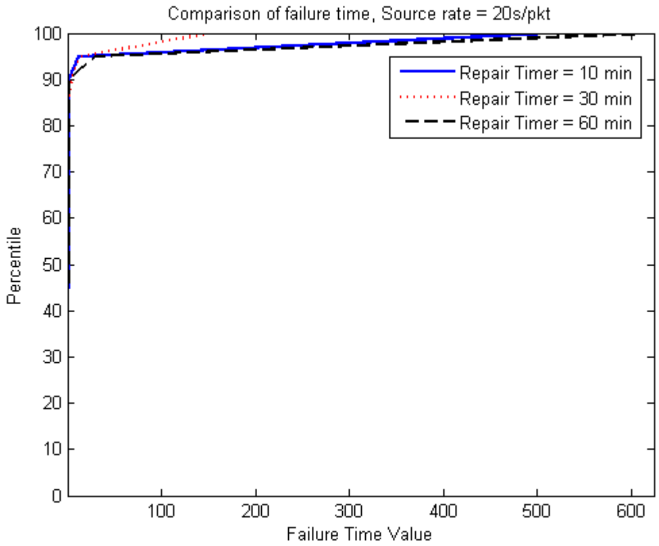

Figure 17: CDF: Loss of connectivity for different DAG repair timer values for gloabal+local repair, packet rate 20/s.

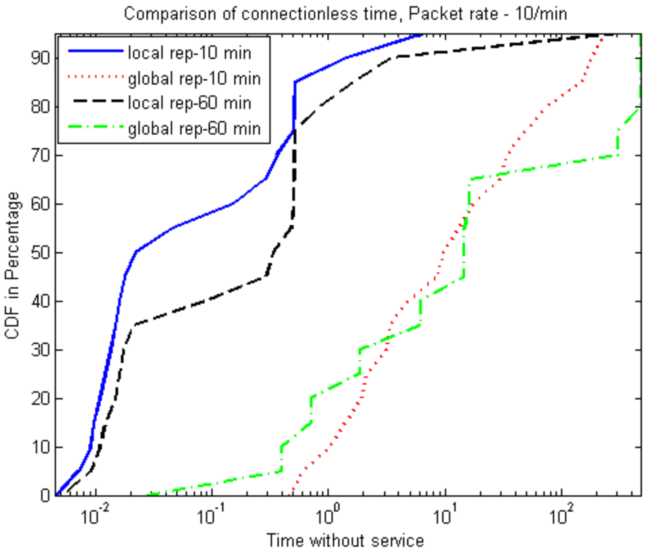

Figure 18: CDF: Comparing loss of connectivity for global repair and global+local repair, packet rate 10/s.

A comparison between the amount of control overhead used for global repair only and global plus local Repair mechanism is shown in Figure 20, which highlights the improved performance of RPL in terms of convergence time at very little extra overhead. From Figure 19, in 85% of the cases the protocol finds connectivity to the LBR for the concerned nodes within fraction of seconds when local repair is employed.

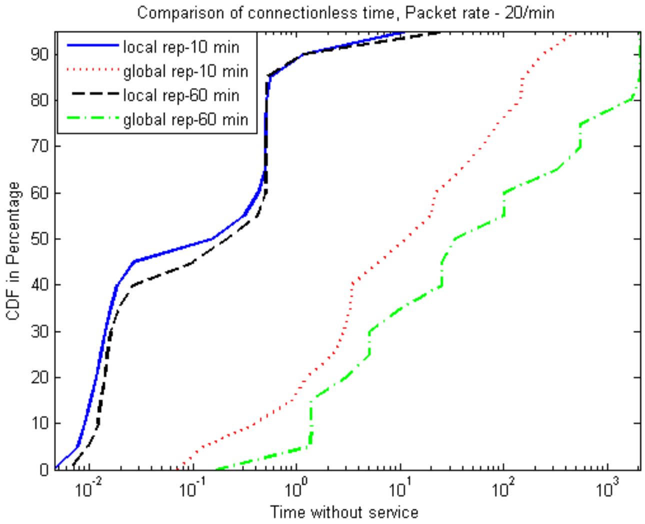

Figure 19: CDF: Comparing loss of connectivity for global repair and global+local repair, packet rate 20/s.

Using only global repair leads to 150 − 154 seconds as observed in Figures 13 and 14.

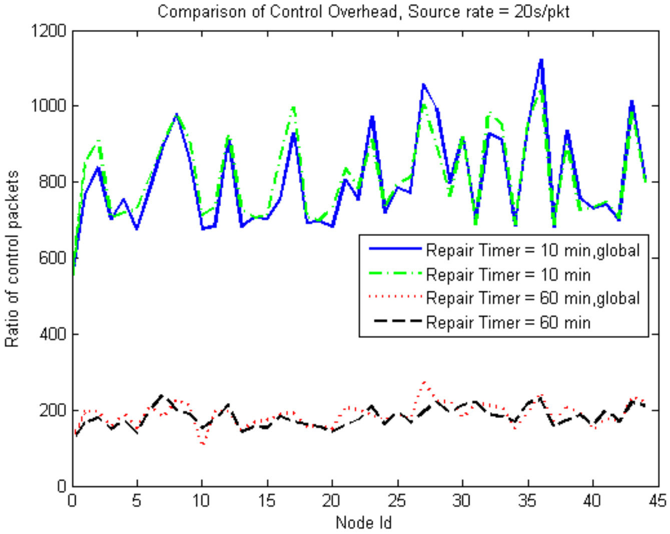

Figure 20: Number of control packets for different DAG Seq Number period, for both global repair and poisoning.

5 RPL in a building automation routing scenario

Unlike the previous traffic pattern, where a majority of the total traffic generated by any node is destined to the root, this section considers a different traffic pattern, which is more prominent in home or building routing scenario. In the simulations shown below, the nodes send 60% of their total generated traffic to the physically 1-hop distant node, 20% of traffic to a 2-hop distant node (again this is a typical traffic pattern in building and home automation networks). The other 20% of traffic is distributed among other nodes in the network. The CDF of average hop distance path stretch in terms of hop distance, ETX path cost and delay for P2P routing for all pair of nodes is calculated. Maintaining low delay bound for P2P traffic is of high importance in this traffic scenario, as the applications in home and building routing have typically low delay tolerance.

5.1 Path Quality

Figure 21 shows the CDF of number of hops for both RPL and ideal shortest path routing for the traffic pattern described above. Figure 22 shows CDF of the expected number of transmission (ETX) for each packet to reach destination. Figures 23 and 24 show CDF of the stretch factor for these two metrics. To illustrate the stretch factor, an example from figure 24 will be given next. For all paths built by RPL, 85% of the time the path cost is less than the path cost for the ideal shortest path plus one.

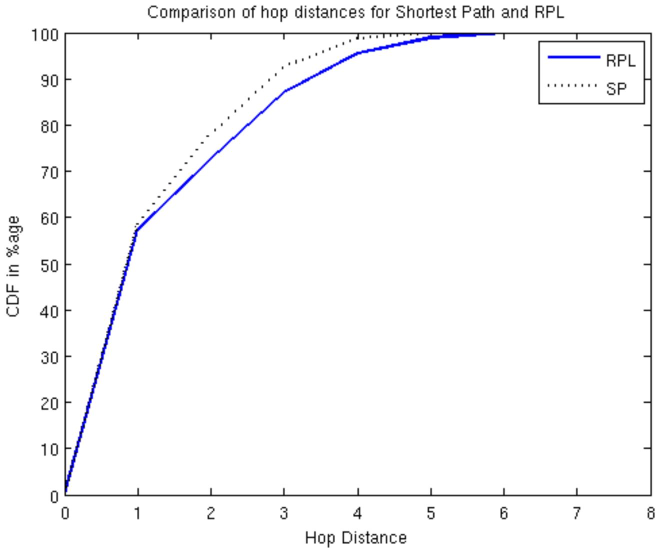

Figure 21: Comparison of end-to-end hop distance for RPL and ideal shortest path in home routing.

5.2 Delay

To get an idea of maximum observable delay in the mentioned traffic pattern, the delay for different number of hops to the destination for RPL is considered. Figure 25 shows how the end-to-end packet latency is distributed for different packets with different hop counts in the network.

To get an idea of maximum observable delay in the mentioned traffic pattern, the delay for different number of hops to the destination for RPL is considered. Figure 25 shows how the end-to-end packet latency is distributed for different packets with different hop counts in the network.

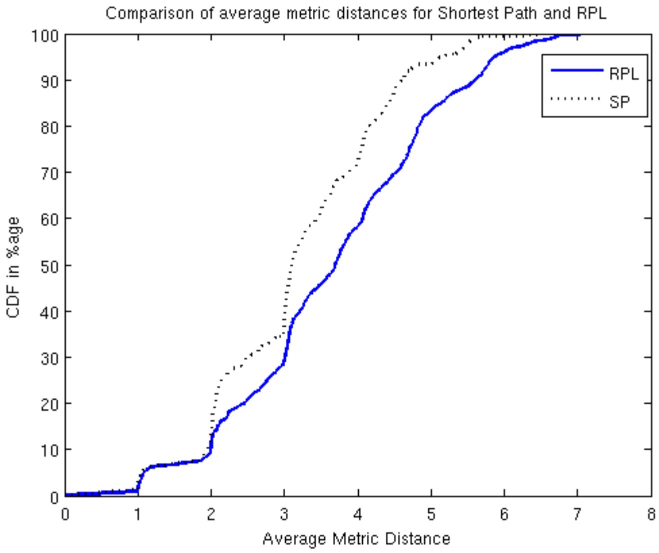

Figure 22: Comparison of path ETX metric for RPL and ideal shortest path in home routing.

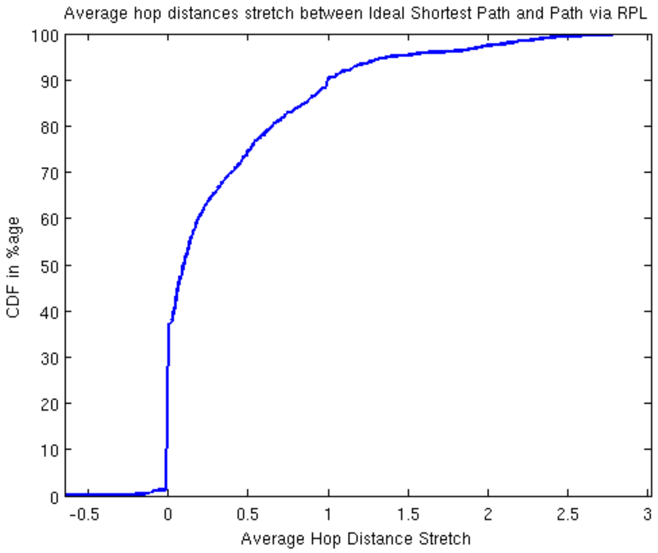

Figure 23: Stretch factor for node hop distance from ideal shortest path.

For this deployment scenario, 60% of the traffic has been restricted to 1-hop neighborhood. Hence, intuitively, the protocol is expected to yield path qualities which are close to that of ideal shortest path routing for most of the paths. From the CDF of hop distance and ETX path cost, it is clear that peer-to-peer paths are more often closer to an ideal shortest path. The end-to-end delay for distances within 2 hops are less than 60 ms for 99% of the delivered packets, while packets traversing 5 hops and more are delivered within 100 ms 99% of the time. These results demonstrate that, for a normal routing scenario of an LLN deployment in a building, RPL performs fairly well without incurring much control plane overhead, and it can be applied for delay critical applications as well.

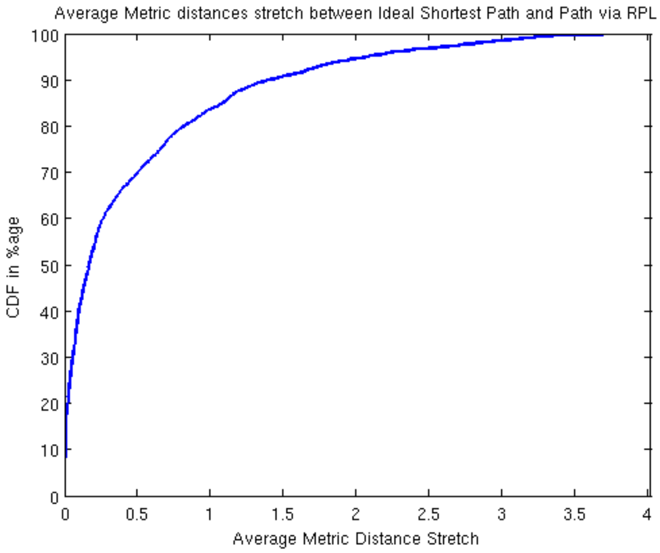

Figure 24: Stretch Factor of total path ETX Metric from ideal shortest path.

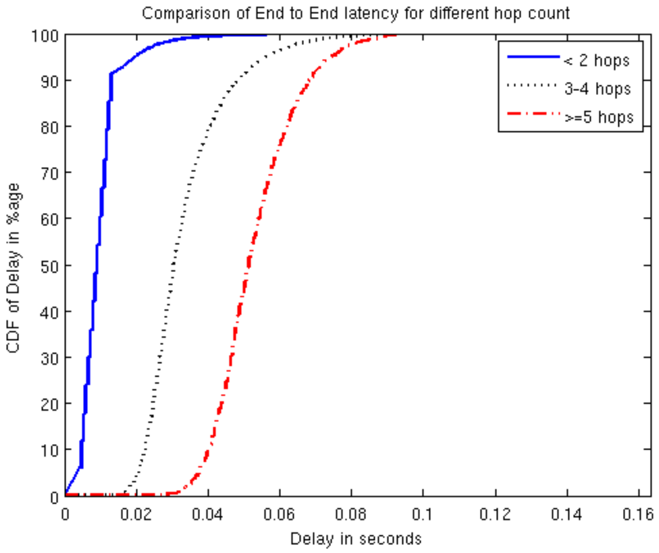

Figure 25: Comparison of packet latency for different hop count in RPL.

6 RPL in a Large Scale Network

In this section we focus on simulating how RPL operates in a large network and study its scalability by focusing on a few performance metrics: the latency and path cost stretch, and the amount of control packets for scalability. The 2442 node smart meter network with its corresponding link traces is used in this scalability study. To simulate a more realistic scenario for a smart meter network, 100% of the generated packets by each node are destined to the root. Therefore, no traffic is destined to nodes other than the root.

6.1 Path Quality

To investigate RPL's scalability, the CDF of ETX path cost in the large scale smart meter network is compared to a hypothetical ideal shortest path routing protocol which minimizes the total path ETX (Figure 26). In this simulation, the path stretch is also calculated for each packet that traverses the network. The path stretch is determined as the difference between the path ETX taken by a packet while following a route built via RPL and a path computed using an ideal shortest path routing protocol. Here, the CDF of fractional path stretch, which is determined as the path stretch value over the path cost of an ideal shortest path, is plotted in Figure 27. The same fractional path stretch value for hop distance is shown in Figure 28.

Looking at the path quality plots, it is obvious that RPL works in a non-optimal fashion in this deployment scenario as well. However, on average, for each source-destination pair, the fractional stretch is limited to 30% of the ideal shortest path cost. This fraction is higher for paths with shorter distance, and lower for paths where source-destination are far apart. The negative stretch factor for hop count is an interesting feature of this deployment and is due to RPL's decision of not switching to another parent where the improvement in path quality is not significant. As mentioned, in this implementation, any node will not switch to a new parent unless the advertised path cost to LBR through the new candidate parent is 20% better the old one. Hence, there are paths where the ETX path cost to LBR is small for a path with a larger number of hops. The nodes tend to hear DIOs from a smaller hop count first, and later do not always shift to a larger hop count and smaller ETX path cost. As the traffic is mostly to the DAG root, some P2P paths built via RPL do yield a smaller hop count from source to destination, albeit a larger ETX path cost.

As observed in Figure 26, 90% of the packets transmitted during the simulation have a (shortest) path cost to destination less than or equal to 12. However, via RPL, 90% of the packets will follow paths that have a total ETX path cost of up to 14. Though all packets are destined to the LBR, it is to be noted that this implementation ignores a change of up to 20% in total path cost. Figures 27 and 28 indicate all paths have a very low fractional stretch factor as total path cost ETX is concerned, and some of the paths have lesser hop counts to LBR as well when compared to the hop count of ideal shortest path.

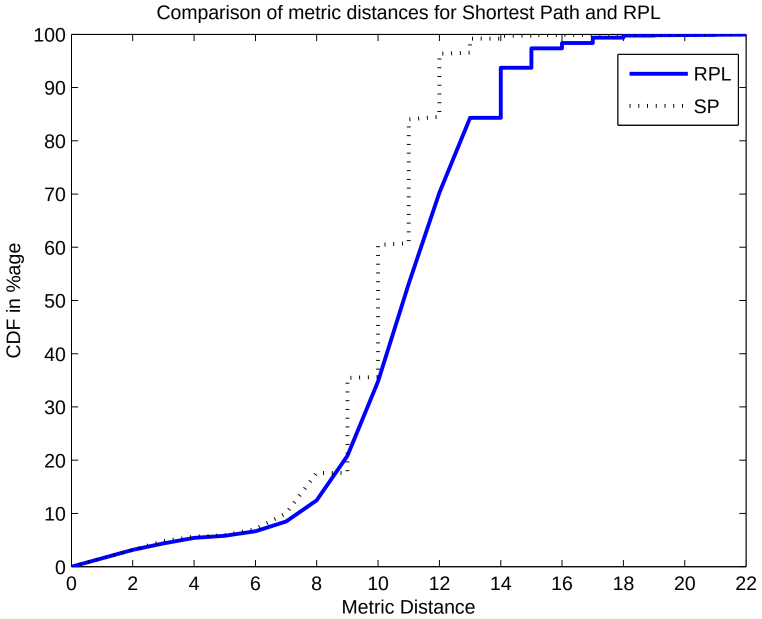

Figure 26: CDF of total ETX path cost

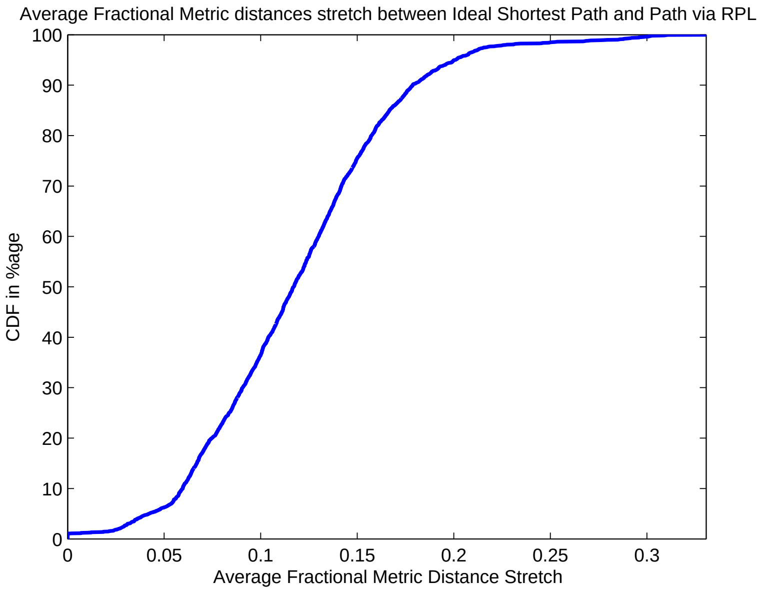

Figure 27: CDF of fractional stretch in ETX path cost

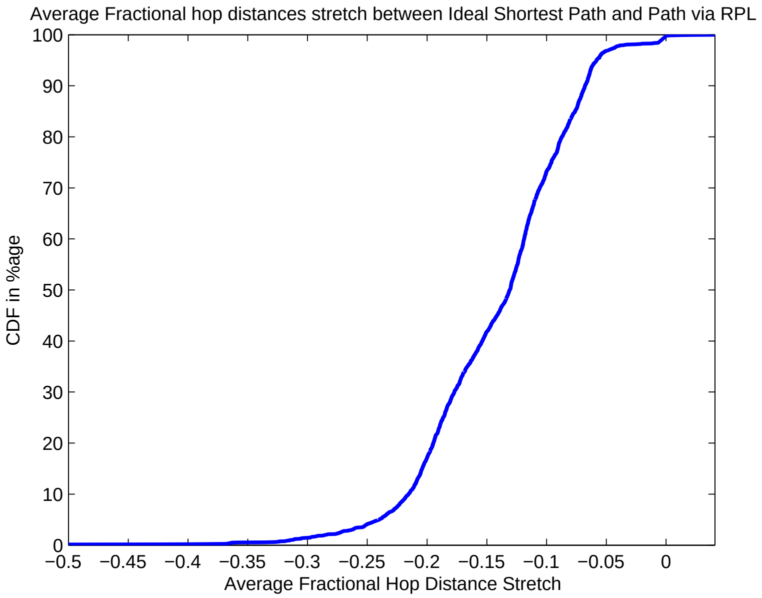

Figure 28: CDF of fractional stretch in hop count

6.2 Delay

Figure 29 shows how the end-to-end packet latency is distributed for different hop counts in the network. According to [RFC5826], U-LLNs are delay tolerant, and the information, except for critical alarms, should arrive within a fraction of the reporting interval (within a few seconds). The packet generation for this deployment has been set higher than usual to incur high traffic volume, and nodes generate data once every 30 seconds. However, the end-to-end latency for most of the packets is condensed between 500 ms to 1s, where the upper limit corresponds to packets traversing longer (larger than or equal to 6 hops) paths.

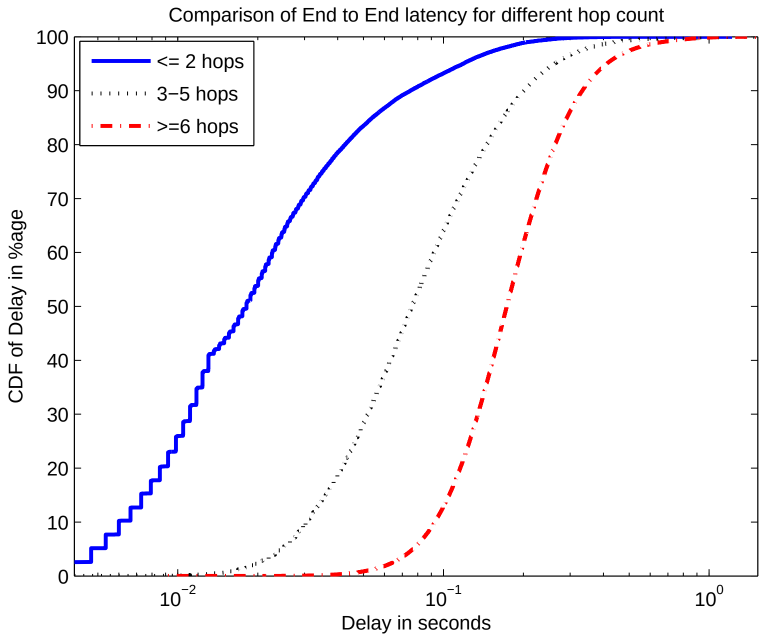

Figure 29: End-to-end packet delivery latency for different hop count

6.3 Control Packet Overhead

Figure 30 shows the comparison between data packets (originated and forwarded) and control packets (DIO and DAO messages) transmitted by each node (link ETX is used as the routing metric). Here one can observe that in spite of the large scale of the network, the amount of control traffic in the protocol is negligible in comparison to data packet transmission. The smaller node id for this network actually indicates closer proximity to the sink and nodes with high ID are actually further away from the sink. Also, as expected, we can observe on Figures 31, 32, 33 that the (non-leaf) nodes closer to the sink have much more data packet transmission than other nodes. The leaf nodes have comparable amount of data and control packet transmission, as they do not take part in routing the data. As seen before, the data traffic for a child node has much less variation than the nodes which are closer to the sink. This variation decreases with increase in DAG depth. In this topology, Nodes 1, 2, and 3, etc., are direct children of the LBR.

Figure 30 shows the comparison between data packets (originated and forwarded) and control packets (DIO and DAO messages) transmitted by each node (link ETX is used as the routing metric). Here one can observe that in spite of the large scale of the network, the amount of control traffic in the protocol is really negligible in comparison to data packet transmission. The smaller node id for this network actually indicates closer proximity to the sink and nodes with high ID are actually further away from the sink. Also, as expected, we can observe on Figures 31, 32 and 33 that the (non-leaf) nodes closer to the sink have much more data packet transmission than other nodes. The leaf nodes have comparable amount of data and control packet transmission, as they do not take part in routing the data. As seen before, the data traffic for a child node has much less variation than the nodes which are closer to the sink. This variation decreases with increase in DAG depth. In this topology, nodes 1, 2 and 3, etc. are direct children of the LBR.

In Figure 34, the effect of global repair period timer on control packet overhead is shown.

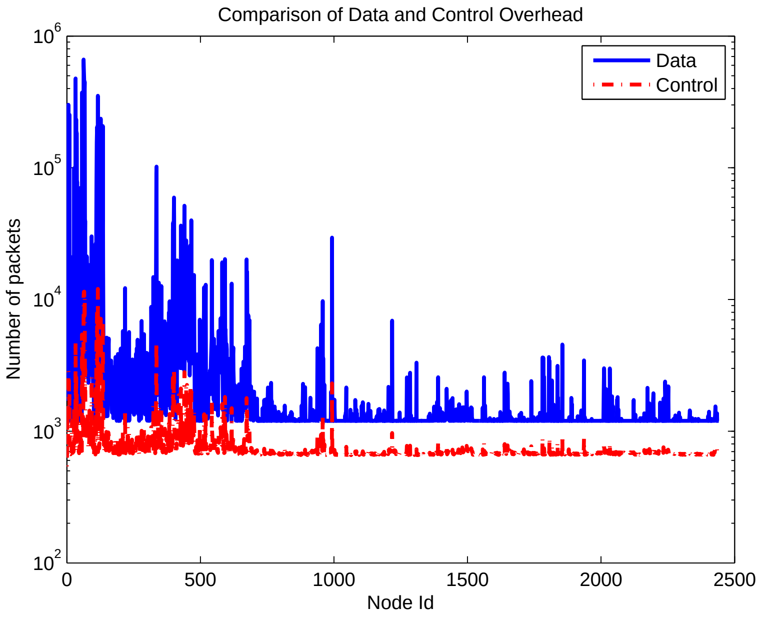

Figure 30: Data and control packet comparison

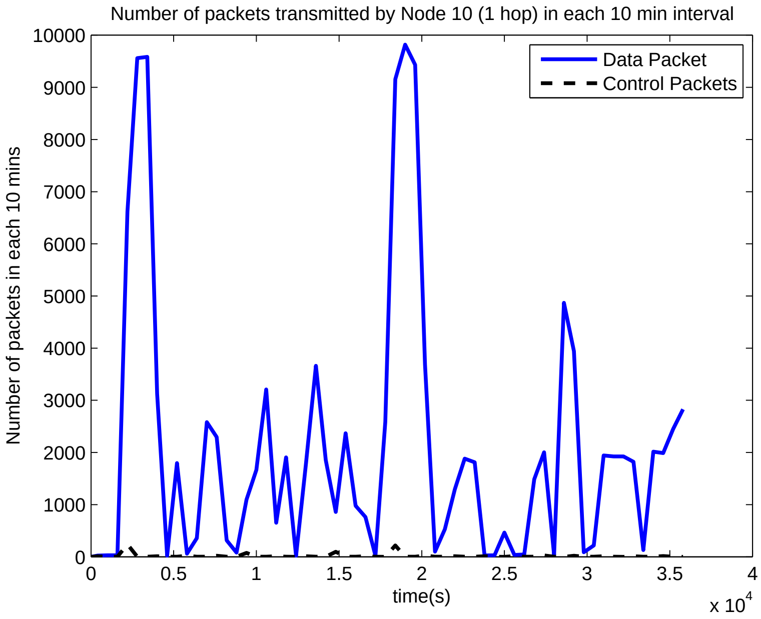

Figure 31: Data and control packet over time for Node 1

7 Scaling Property and Routing Stability

An important metric of interest is the maximum load experienced by any node (CPU usage) in terms of the number of control packets transmitted by the node. Also, to get an idea of scaling properties of RPL in large scale networks, it is also key to analyze the number of packets handled by the RPL nodes for different sizes of the network.

In these simulations, at any given interval, the node with maximum control overhead load is identified. The amount of maximum control overhead processed by that node is plotted against time for three different

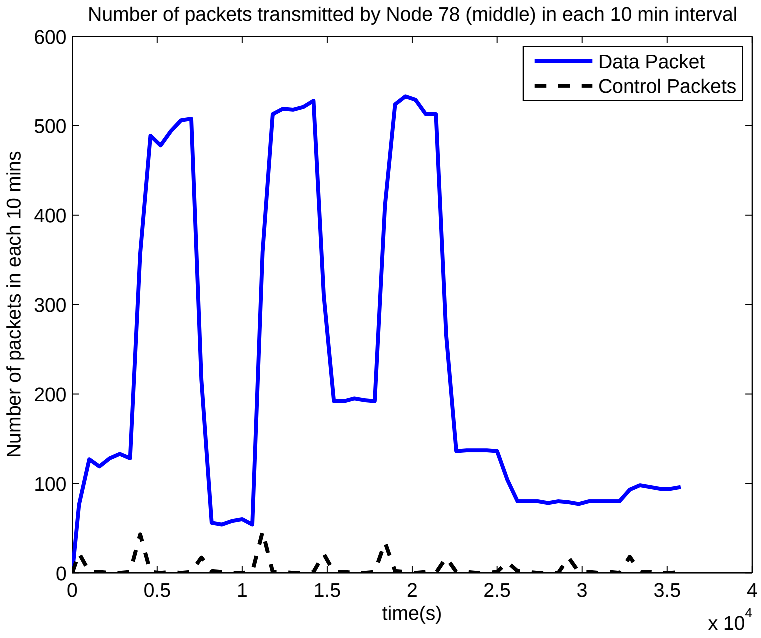

Figure 32: Data and control packet over time for Node 78

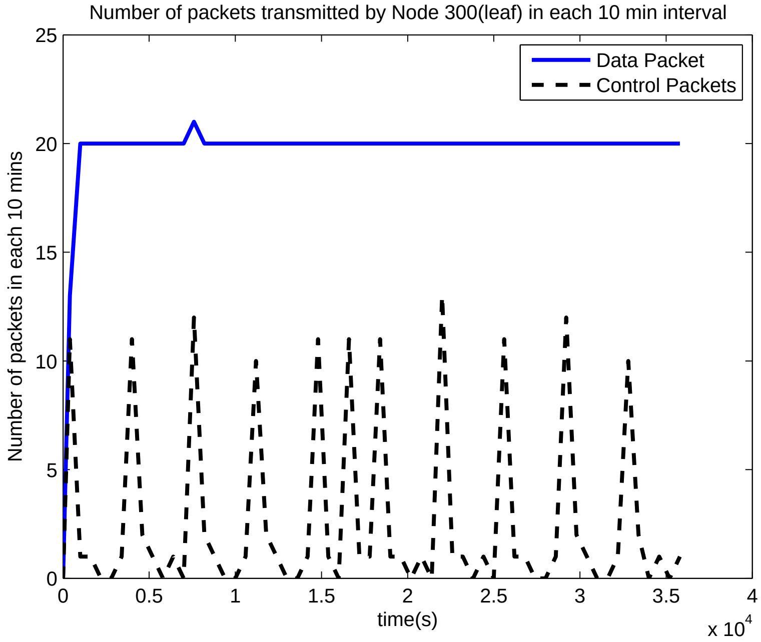

Figure 33: Data and control packet over time for Node 300

networks under study. The first one is Network ‘A’, which has 45 nodes and is shown in figure 1 in section 3; Network ‘B’, which is another deployed outdoor network with 86 nodes; and finally, Network ‘C’, which is the large deployed smart meter network with 2442 nodes being considered in this document.

In Figure 35, the comparison of maximum control load is shown for different network sizes. For the network with 45 nodes, the maximum number of control packets in the network stays within a limit of 50 packets (per 1 minute interval), where for the networks with 86 and 2442 nodes, this limit stretches to 100

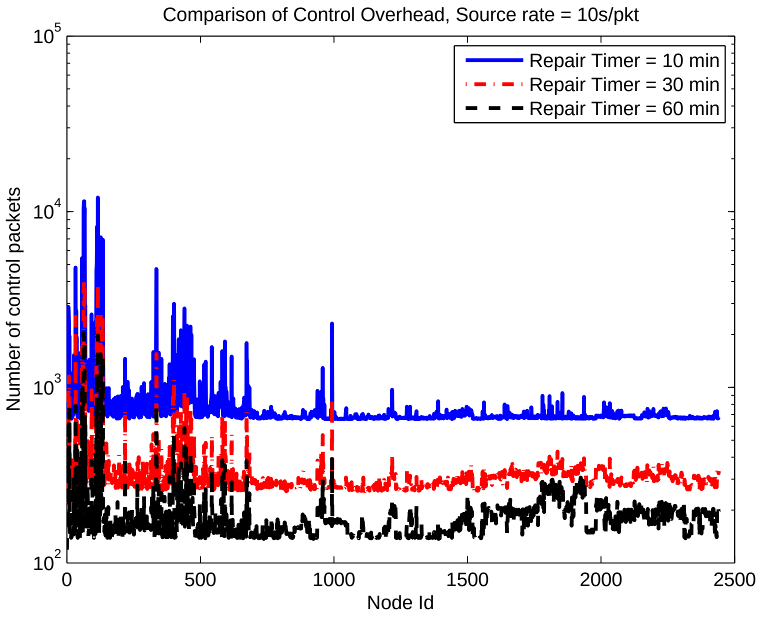

Figure 34: Amount of control packet for different global repair timer period

and $2 * 10^3$ packets per 1 minute interval, respectively.

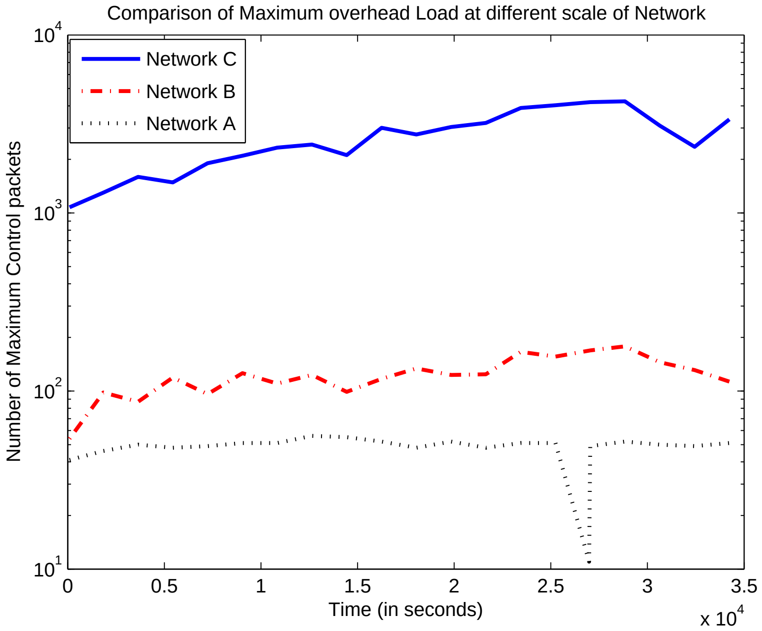

Figure 35: Scaling property of maximum control packets processed by any node over time.

For a network built with low power devices interconnected by lossy links, it is of the utmost importance to ensure that routing packets are not flooded in the entire network, and that the routing topology stays as stable as possible. Any change in routing information, specially parent-child relationship, would reset the timer leading to emitting new DIOs, and hence, change the node's path metric to reach the root. This change will trigger a series of control place messages (RPL packets) in the DODAG. Therefore, it is important to carefully control the triggering of DIO control packets via the use of thresholds.

In this study, the effect of the tolerance value before emitting new DIO reflecting a new path cost is analyzed. Four cases are considered in this study:

- No change in DAG depth of a node is ignored.
- The implementation ignores 10% of change in the ETX path cost to the root. That is, if the change in total path cost to LBR, due to a DIO reception from most preferred parent or due to shifting to another parent, is less than 10%, the node will not advertise the new metric to the root,
- The implementation ignores 20% change in the path cost to the root for any node before deciding to advertise a new depth, and
- The implementation ignores 30% change in the path cost to the root for any node before deciding to advertise a new depth.

This decision does affect the optimum path quality to the root. As observed in Figure 36, for 0% tolerance, 95% of paths used have a stretch factor less than 10%. Similarly, for 10% and 20% tolerance level, 95% of paths will have a 15% and 20% fractional path stretch. However, the increased routing stability and decreased control overhead is the profit gained from the 10% extra increase in path length or ETX, whichever is used as the metric to optimize DAG.

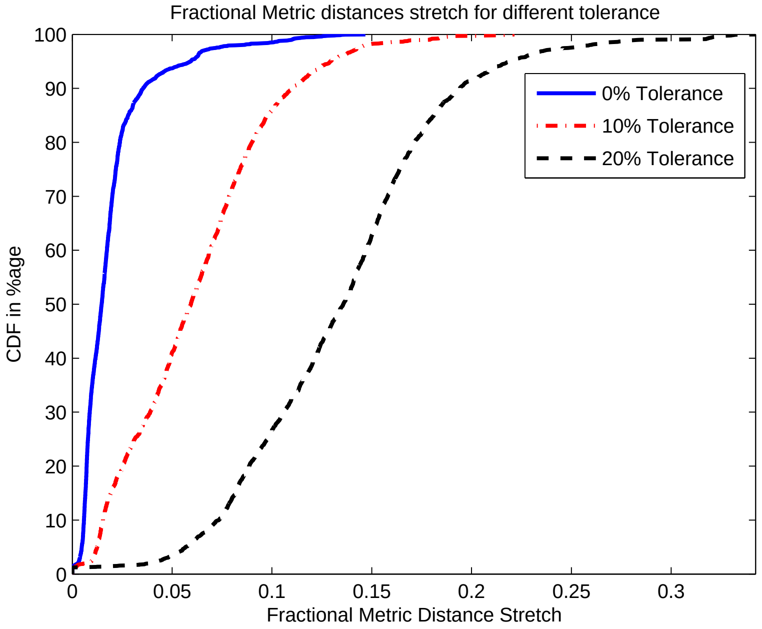

Figure 36: Fractional stretch factor for different tolerance levels

As the above mentioned threshold also affects the path taken by a packet, this study also demonstrates the effect of the threshold on routing stability (number of times P2P paths change between a source and a destination). For Network 'A' shown in Figure 1 in section 3 and the large smart meter network 'C', the CDF of path change is plotted against fraction of path change for different thresholds triggering the emission of a new DIO upon path cost change.

In Figures 37 and 38, we show the CDF of fraction of times a path has changed (for each source-destination pair). If X packets are transferred from source A to destination B, and out of X times, Y times the path between this source-destination pair is changed, then we compute the fraction of path change as

$Y/X * 100\%$. This metric is computed over all source-destination pairs, and the CDF is plotted in the Y axis.

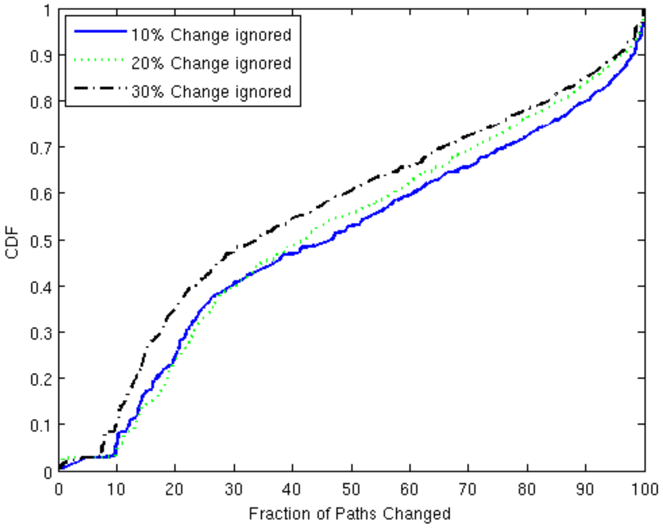

Figure 37: Distribution of fraction of path change, Network 'A'

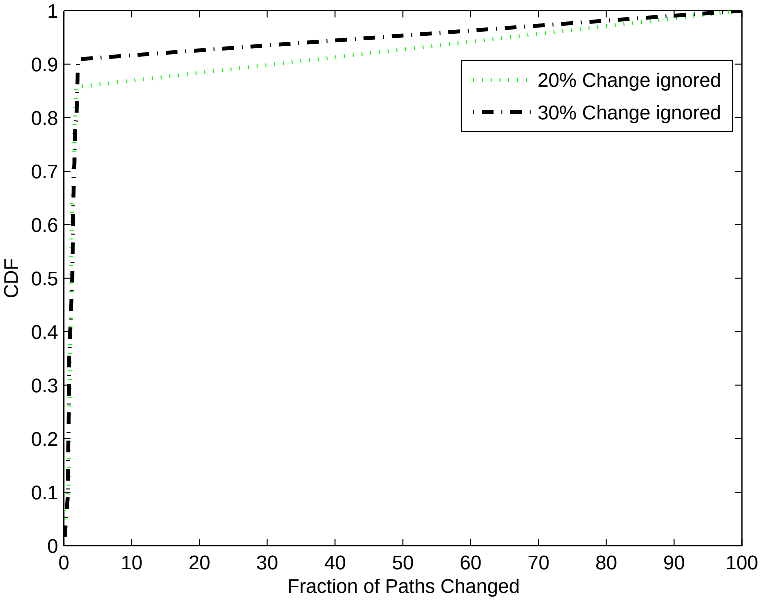

Figure 38: Distribution of fraction of path change, large Network 'C'

This document also compares the CDF of fraction of path change for three different networks, 'A', 'B' and 'C'. Figure 39 shows how the three networks exhibit change of P2P path when 30% change in metric cost to the root is ignored before shifting to a new parent.

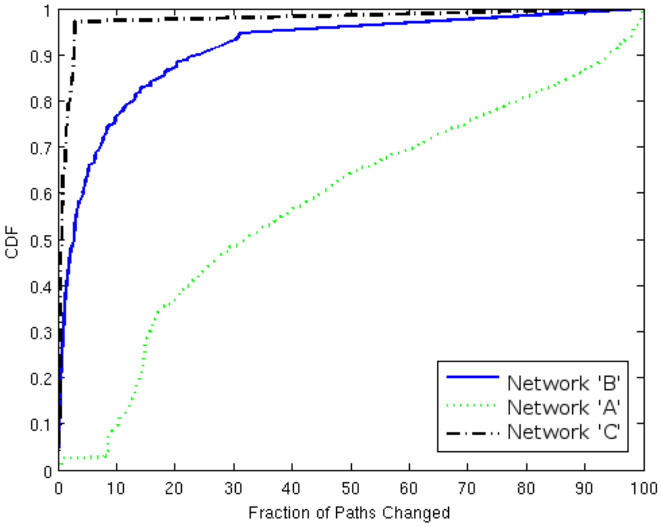

Figure 39: Comparison of distribution of fraction of path change

8 Comments

All the simulation results presented in this document point that the protocol does behave as expected for the particular target scenarios. For the discussed scenarios, the protocol is shown to meet the desired delay and convergence requirements and to exhibit self healing without external intervention, incurring negligible control overhead (only a small fraction of data traffic). RPL also provided path quality which is close to optimum or ideal shortest path for most of the packets in the scenarios considered and is able to trade-off control overhead for path quality as per the application and device requirement through configurable parameters (such as decision on when to switch to new parent), and thus can trade-off routing stability for control overhead as well. Finally, as per the requirement of Urban LLN deployments, the protocol is shown to scale to larger topologies (few thousand nodes).

9 Acknowledgements

The authors would like to acknowledge Jerald P. Martocci, Mukul Goyal, Emmanuel Monnerie, Philip Levis, Omprakash Gnawali and Craig Partridge for their valuable and helpful suggestions over metrics to include and overall feedback.

10 References

10.1 Normative References

[RFC2119] Bradner, S.,“Key words for use in RFCs to Indicate Requirement Levels”, BCP 14, RFC 2119, March 1997.

10.2 Informative References

[RFC5867] Martocci, J., Riou, N., Mil, P., and W. Vermeylen, “Building Automation Routing Requirements in Low Power and Lossy Networks”, June2010.

[RFC5826] Brandt, A., Buron, J., and G. Porcu, “Home Automation Routing Requirements in Low Power and Lossy Networks”, April 2010.

[RFC5673] Pister, K., Thubert, P., Dwars, S., Phinney, T., “Industrial Routing Requirements in Low Power and Lossy Networks,”, October 2009.

[I-D.ietf-roll-routing-metrics] Vasseur, JP, Kim, M., Pister, K., Chong, H., “Routing Metrics used for Path Calculation in Low Power and Lossy Networks, draft-ietf-roll-routing-metrics-19 (work in progress)”, March 2011.

[I-D.ietf-roll-rpl] Winter, T., Thubert, P., et al., “RPL: Routing Protocol for Low Power and Lossy Networks, draft-ietf-roll-rpl-04 (work in progress)”, March 2011.

[I-D.ietf-roll-terminology] JP Vasseur, “Terminology in Low power And Lossy Networks, draft-ietf-roll-terminology-05 (work in progress)”, March 2011.

[NS-2] “The Network Simulator-2”.

[OMNETpp] A. Varga, “The OMNeT++ Discrete Event Simulation System”, June 2001.

[RFC5548] Dohler, M., Watteyne, T., Winter, T., and D. Barthel, “Routing Requirements for Urban Low-Power and Lossy Networks”, RFC 5548, May 2009.

[draft-iphc] J. Jurski, “Limited IP Header Compression over PPP, draft-jurski-pppext-iphc-02.txt (work in progress)”, March 2007.

Authors’ Addresses

Joydeep Tripathi (editor) Drexel University
3141 Chestnut Street 7-313
Philadelphia, PA 19104
USA

Email: jt369@drexel.edu

Jaudelice C. de Oliveira (editor)
Drexel University
3141 Chestnut Street 7-313
Philadelphia, PA 19104
USA

Email: jau@ece.drexel.edu

JP Vasseur (editor)
Cisco Systems, Inc.
1414 Massachussetts Avenue
Boxborough, MA 01719
USA

Email: jpv@cisco.com