A Performance Evaluation Study of RPL: Routing Protocol for Low Power and Lossy Networks

J. Tripathi, J. C. de Oliveira, J. P. Vasseur
Electrical & Computer Engineering Department
Drexel University, Philadelphia, PA 19104; Cisco Systems.
{jt369@,jau@ece.}drexel.edu, jpv@cisco.com

Abstract—**In this paper, a performance evaluation of the Routing Protocol for Low power and Lossy Networks (RPL) is presented. Detailed simulations are carried out to produce several routing performance metrics using a set of real-life scenarios. A real outdoor network was reproduced in simulation with the help of topology and link quality data. Behaviors of such a network in presence of RPL, their reason and potential areas of further study/improvement are pointed out.**

I. Introduction

Designing routing protocols in low power devices in networks with lossy links imposes great challenges, mainly due to low data rates and processing resources, and strict energy constraint in the nodes, as well as high probability of packet delivery failure. While routing in sensor networks has been researched extensively, most commercial solutions are proprietary and suffer from lack of interoperability and therefore would require the use of protocol translating gateways or proxy elements. The IETF Routing Over Low power and Lossy networks (ROLL) Working Group was formed to design a routing solution for low power and lossy networks (also known as sensor networks). Existing routing protocols such as OSPF, IS-IS, AODV, and OLSR have been extensively evaluated by the working group and have been found to not satisfy, in their current form, all of specific routing requirements or Low power and Lossy Networks (LLNs) [1]. Therefore, the working group has specified the Routing Protocol for Low power and Lossy Networks (RPL) in [2]. RPL is designed to meet the core requirements specified in [3], [4], [5] and [6].

This paper's contribution is to provide several routing performance metrics of RPL using a discrete event simulator in real-life deployment scenarios. The RPL protocol is being developed as a standard to be deployed in a number of environments: urban networks, smart grid networks, industrial networks, building and home networks. To the best of our knowledge, no comprehensive efforts to document its performance in real life scenario had been carried out. The understading of RPL's performance in such scenarios is of very much interest to the working group and the community. In this paper, results from a medium scale outdoor network deployment are presented. Several routing metrics are evaluated:

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

Although simulation cannot formally prove that a protocol operates properly in all situations, it could 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 especially when theoretical model assumptions may not be applicable to such networks and scenarios. Therefore, real deployed network data traces have been used to model link behaviors.

The rest of this paper is organized as follows. In Section II, a brief overview of how RPL operates is sketched. Section III provides details of the deployed network from where we gather the link trace and topology. Section IV describes how simulation was carried on, and Section V documents the observation on the performance metrics.

II. RPL: Routing Protocol for LLNs

The Working Group focusses in the use of RPL in conjunction with IPv6 routing architectural framework as they are suitable for time varying link loss characteristics and have low CPU and memory requirements [1]. The motivation of RPL is to construct a Directed Acyclic Graph (DAG) rooted at the sink, which will minimize the cost of reaching the sink from any node in the network as per the Objective Function (OF). The OF can be to minimize a particular metric, such as hop count or ETX (Expected Transmission count). A list of metrics is specified in [7]. Considering the vast number of applications for RPL, the routing protocol has been designed with a great deal of flexibility and supports a variety of OFs in order to build the routing topology according to various link/node metrics and constraints. RPL is an IPv6 distance vector routing protocol that builds Destination Oriented Directed Acyclic Graph (DODAG). The root of the DAG starts sending out DAG Information Option (DIO) messages. The DIO messages contain the information about the rank of the broadcasting node (which is basically the distance of the node from the backbone network), the OF, DAG-ID, etc. Any node that is connected to a non-RPL implementing node or a backbone network, can act as a root or LBR (LLN Border Router) and has a rank equal to 1. Once a node receives a DIO, it calculates its rank based on the rank in the received DIO, and the cost of

978-1-4244-7417-2/10/$26.00

reaching the node from itself. RPL defines a number of rules for parent selection based on the local quality of the links, the advertised OF, path cost, rank, etc. and any node that has lower rank than the node itself is considered as a candidate parent. When a node broadcasts its DIO, it includes all information about its rank, OF and the DAG it has joined in. This way, DIOs propagate down to most distant nodes from root and help create a DODAG in the network. DIOs are emitted periodically from each node, triggered by a timer (trickle timer) whose duration increases exponentially (doubled after each time it is fired). The smallest possible interval between two DIOs is denoted by I_min, and number of times I_min can be doubled before maintaining a constant rate is denoted by I_doubling, so $I\_max = I\_min * 2^{I\_doubling}$. On any event that causes a change in the DAG structure (parent node unreachable, new parent selection, new DAG Squence Number etc.), this timer is reset to the I_min mentioned in DIOs.

When a node joins the network, it may wait to receive a DIO or it may alternatively multicast a solicitation message called the DIS, so that other nodes hearing the DIS starts sending DIOs, and the newly arrived node can join the DAG. Nodes also multicast Destination Advertisement Option or DAOs for 1-hop reachability and unicast reachable prefix to their parents in the DAG to advertise their addresses and prefixes. The nodes that receive these DAOs, update their routing table. When no entry is available in the routing table, or for traffic to the root, a node will forward a packet up to its most preferred parent. Note that while sending packets up the DAG, a node must not forward it to a node with greater rank to prevent a loop in the routing path. In case no parent is available, the node can forward the data to a sibling (node with same rank). This way, for P2P routing, the packet goes from the source up to a common ancestor of the source and destination in the DAG, and then it travels down to the destination. The LBR will periodically emit DIO with new sequence number (DAGSequenceNumber) in order to recalculate the DAG and repair any broken link. When a node encounters a DIO with greater sequence number, it again starts the parent selection process as per the updated link cost and confidence about the parents, so for the next iteration it has a better path to the root. It also helps a node find a parent in case their parent list is depleted due to unsuccessful probes or low confidence.

III. Network Setup

Real link layer data gathered from networks deployed on the field were used to compute the PDR (Packet Delivery Ratio) for each of the links in the network. By contrast with theoretical models (e.g. Markov Chains) which may have assumptions not applicable to lossy links, real-life data has been used for two aspects of the simulations:

- **Topology:** The topologies are gathered from real-life deployment (traces mentioned above) as opposed to random topology simulations. A 86 node topology with a single sink/root, shown in 1, gathered from a real outdoor deployment, was used in the simulations. The links with color red are part of the DAG, i.e. between a node and its most preferred parent. Note that this is just a start to validate the simulations before using large scale networks.

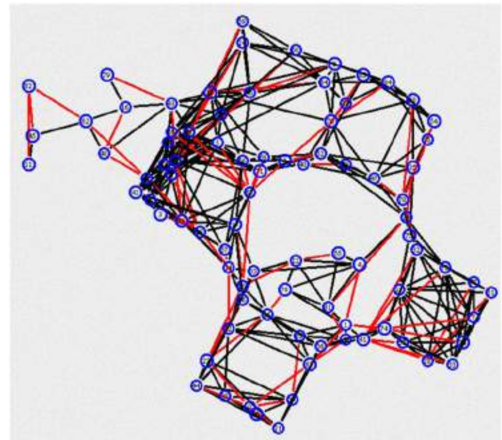

Fig. 1. Network topology for preliminary simulation results.

- **Link Failure Model:** A database of time varying link quality data, gathered from the same real outdoor network deployment, was created. Each link in the topology 'picks up' the same link model from the database of gathered trace, and the link's PDR varies according to the real scenario with time (in simulation, new PDR is read from the database every 10 minutes). Packets are dropped randomly from that link with probability (1 - PDR). Each link uses a different Mersenne Twister 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 such as the DIO, DAO, DIS packets defined in [2]. Figure 2 shows a typical temporal characteristic of some links in the outdoor network trace used in the simulation.

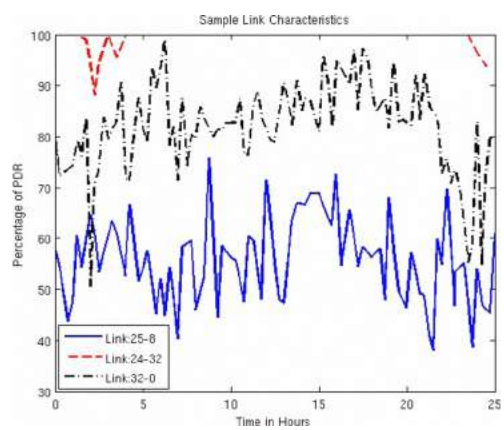

Fig. 2. Example of link characteristics.

IV. Simulation Setup

RPL was simulated using OMNET++ [8], a well-known discrete event based simulator written in C++ and NED. Castalia-

2.2 [9] has been used as Wireless Sensor Network Simulator framework within OMNET++. The output and events in the simulation are visualized, for verification purposes, with the help of the Network AniMator or NAM, which is distributed with NS (Network Simulator) [10]. Note that NS or any of its versions were not used for the simulation itself. The radio was simulated as TelosB CC2420 radio, and 802.15.4 MAC protocol was used.

In simulating RPL, the LBR first initiates sending out DIO messages, and the DAG is gradually constructed. The trickle time interval for emitting DIO message assumes the initial value of 1 second, and then changes over simulation time as mentioned in [2]. 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. 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 IETF ROLL Working Group applicability statement documents. Further study for large scale networks with varied parameters can show how quickly the network will stabilize, comparing data/control traffic and studying the trade-off between reactivity and lifetime.

Each node in the network, other than the LBR, also emits DAO messages as specified in [2]. This is done to initially populate the routing tables with the prefixes received from children via the DAO messages in support of the Point to Point (P2P) and Point to Multipoint traffic (P2MP) in the "down" direction. In this study, it is assumed that each node is capable of storing route information for other nodes in the network, therefore no source routing is required.

Each node sends traffic according to a Constant Bit Rate (CBR) to all other nodes in the network over the simulation period. To simulate a more realistic scenario, 20% of the generated packets by each node are destined to the root, and the remaining 80% 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 (more than 20 times). These values may be revised when studying the P2P traffic so as to have a majority of traffic going to all nodes as opposed to the root. The packets are routed through the DAG built by RPL according to the mechanisms specified in [2]. Since RPL is an IPV6 routing protocol, no assumption is made on the link layer, thus potential gains in terms of header compression provided by 6loWPAN is not under consideration [11]. A number of RPL parameters are configured (such as Packet Rate from each source, Time Period of the LBR emitting new DAG sequence number) to observe their effect on the performance metrics of interest.

V. Metrics of Interest and Simulation Results

A. Common Assumptions

Routing Table Size: As the DAO messages help to feed the routing tables in the network, routing table size for each node are recorded. Currently, the routing table size is not in terms of Kbyte of memory usage but measured in terms of number of entries for each node. Each entry has next hop node and path cost associated with the destination node. The ETX (Expected Transmission Count) metric is used to build the DAG as specified in [7].

B. Path Quality

Number of Hops: For each pair of source and destination, the average number of hops for both RPL and shortest path routing is computed. Shortest path routing refers to an hypothetical ideal routing protocol that would always provide the shortest path in terms of Total ETX (or whichever metric is used) in the network. The Cumulative Distribution Function (CDF) of hop distance for all paths (which is equal to n*(n-1) in an n node network) in the network with respect to number of hops is plotted in Figure 3 for both RPL and shortest path routing. One can observe that the CDF corresponding to 5 hops is around 60% for RPL and 90% for shortest path routing. This means, for the given topology, 90% of paths will have path length of 5 hops or less with an ideal shortest path routing methodology, whereas in RPL Point-to-Point (P2P) routing, 90% of paths will have a length shorter or equal to 8 hops. This result shows that despite having a non optimized P2P routing scheme, the path quality of RPL is not drastically worse than ideal. On-going studies actually show that the it is fairly close in both cases. Furthermore a number of optimizations can be added (such as increasing the degree of meshing via the OF) to get even closer in terms of path quality. This result may be different in other topologies. If the sink/root is in the middle of the network, and not at one end, like in our simulation, P2P path quality will be better.

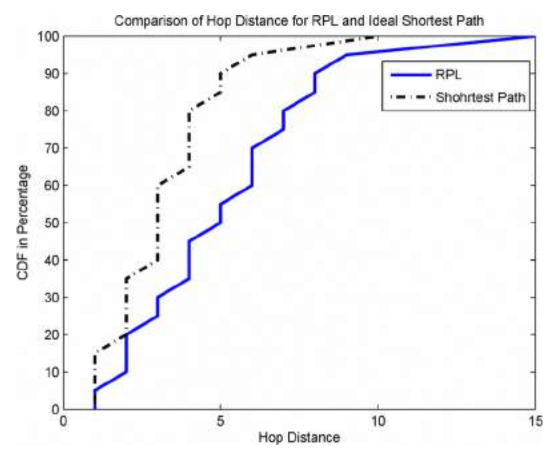

Fig. 3. CDF: hop distance versus number of hops.

Total ETX: When optimizing ETX metric along the path is used as an objective function, the total ETX along the path is computed for each pair. Figure 4 shows the CDF of the total number of ETX to deliver a packet from a source to any destination node with respect to total ETX of the path from each source to each destination for the network, for both RPL, and a hypothetical ideal shortest path mechanism. Here also one observes that total ETX along the path from all source to all destination is close to that of a shortest path for the network in our simulation.

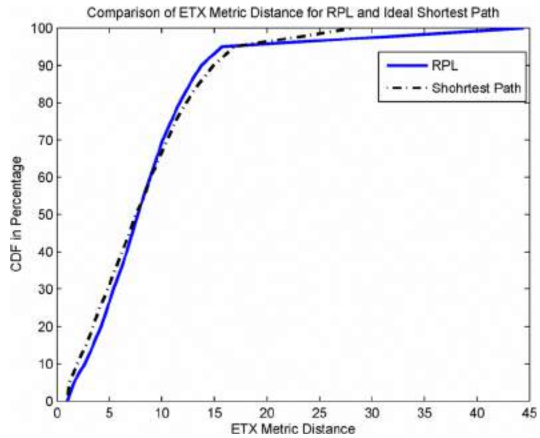

Fig. 4. CDF: Total ETX along path versus ETX value.

C. Routing Table Size

The objective of this metric is to observe the distribution of the the number of entries per node. Figure 5 shows the CDF of required number of routing table entries for all nodes. One can see that 90% of the nodes need to store less than 21 entries in their routing cache.

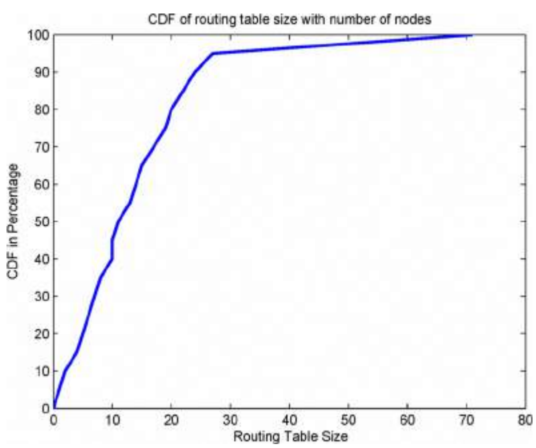

Fig. 5. CDF of routing table size with respect to number of nodes.

D. Control Packet Overhead

The control plane overhead is an important routing protocol characteristics in LLNs considering how scarce the resources are. Indeed, 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 metric is thus to analyse the control plane overhead in stable condition (no network element failure overhead) and in the presence of failures.

Data and control plane traffic comparison for each node: Figure 6 shows the comparison of the amount of data packets transmitted (including forwarded) and control packets (DIO and DAO messages) transmitted for each node when minimizing ETX is used as the OF along the DAG. Here one can observe that considerable amount of traffic is routed through the sink/root itself. And also the fact that the amount of control traffic is really negligible in the protocol is reinforced. As expected, the nodes closer to sink and that act as routers 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.

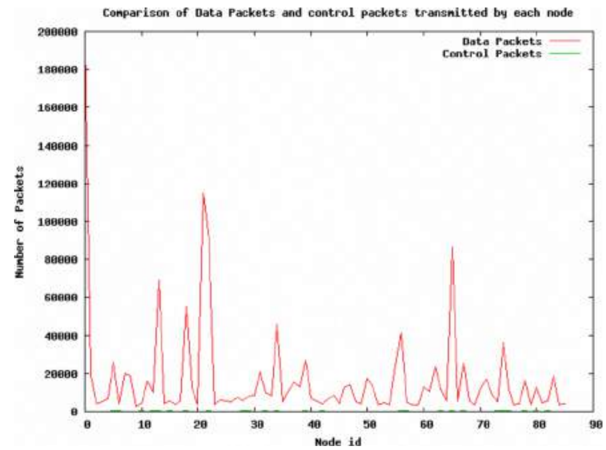

Fig. 6. Amount of data and control packets transmitted for each node

Data and Control Packet Transmission with respect to time: In Figures 7, 8 and 9, the amount of data and control packets transmitted for node 22 (high rank in DAG, a neighbor of sink itself), node 78 (in the middle of the DAG) and node 16 (a leaf node) are shown, respectively. These values stand for number of packets transmitted for each 10 minutes intervals, to help understand what is the density of data and control packet exchange in the network. One can observe as the node is closer to the sink, the amount of data is larger, and the amount of control traffic is negligible in comparison to the data traffic. Also, the variation in data traffic is much larger for a node closer to sink, because the destination of the packets varies over time, and 20% of the packets are destined to sink. For the nodes that are farther away from sink , the variation in data traffic becomes lesser, and the amount of data traffic is also smaller.

The control traffic for the nodes has a wave-like pattern. The amount of control packets for each node drops quickly as the DAG stabilizes due to the effect of trickle timer. However, as a new DAG Sequence is advertised, the trickles are reset and the nodes start emitting DIO frequently again to resolve DAG inconsistencies. Also when the topology changes, RPL dynamically adapts the routing topology leading to increased control traffic until the DAG stabilizes by reseting the trickle timer. The number of control packets attains a high value for one interval, and the amount comes down to lower values for subsequent intervals. One can see that for a node closer to sink, the data packet amount is much higher than control packet, and somewhat oscillatory around a mean value. Also, for leaf nodes the amount of control packets are more than data packets, as leaf nodes are more prone to face changes in their DAG depth as opposed to nodes closer to sink.

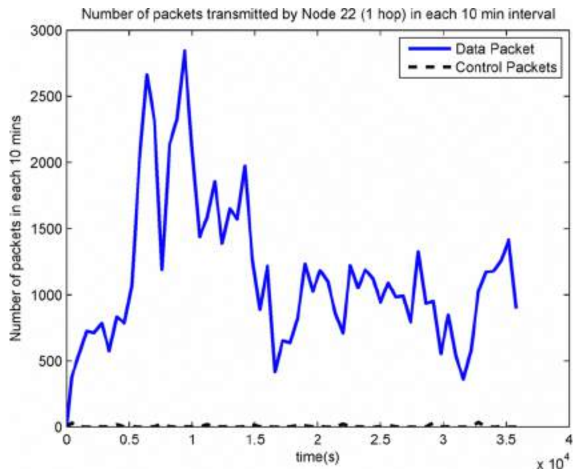

Fig. 7. Amount of data and control packets transmitted for node 22.

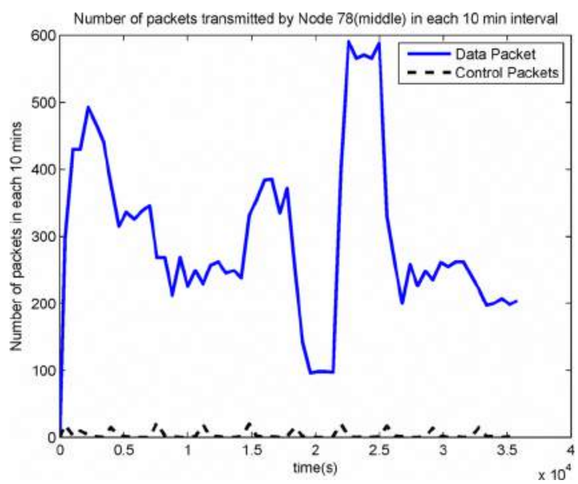

Fig. 8. Amount of data and control packets transmitted for node 78.

E. Loss of connectivity

Upon link failures, a node may loose both his parents (preferred and backup) and its sibling (if any). In this case, if a packet has to be sent and the routing table does not contain an entry for the corresponding destination the packet is dropped. RPL proposes two mechanism for DAG repairs, known as a) global repair, which is implemented with the help of periodic emission of new DAG Sequence Number by the DAG root; and b) local repair, whereby upon loosing parents, a node will try to quickly and locally find an alternate parent. Thanks to local repair, RPL provides shorter convergence time but our objective in this paper is to exclusively rely on global repair due to space limitation. More detail about local repair and detailed operations of RPL in LLNs can be found in the book

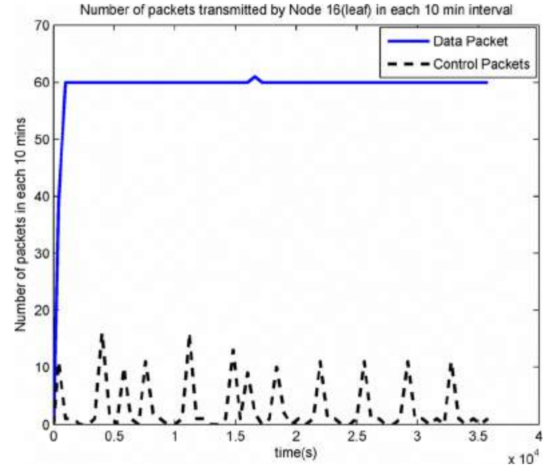

Fig. 9. Amount of data and control packets transmitted for node 16.

in [12]. The idea is to tune the frequency at which new DAG Sequence Numbers are generated by the DAG root that is used for global repair. It is expected that a higher frequency of new Sequence Numbers will lead to shorter duration of connectivity loss at a price of a higher rate of control packet in the network if only Global repair is used.

Figure 10 shows the CDF of time spent by any node without any connectivity to root, i.e. a confident parent to forward data when the packet rate from the sources is a packet each 10 seconds, and new DAG Sequence Number is issued every 10 minutes. This plot reflects the properties of RPL without any Local Repair scheme. When all the parents (and siblings) are temporarily unreachable from a node, the time before it hears a DIO from another node is recorded, which gives the time without service. In some cases, this value might go up to the DAG Repair Timer value, because until a DIO is heard, the link outage is not solved.

The effect of the DAG Repair Timer on time without service is plotted in Figure 11, where the source sends packet every 20 seconds and in Figure 12, where the source sends a packet every 10 seconds.

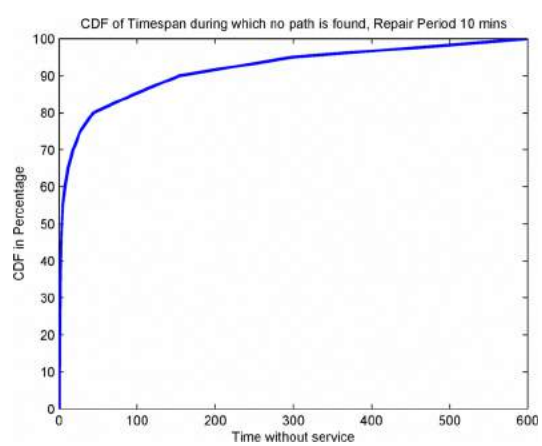

Fig. 10. CDF: Loss of connectivity.

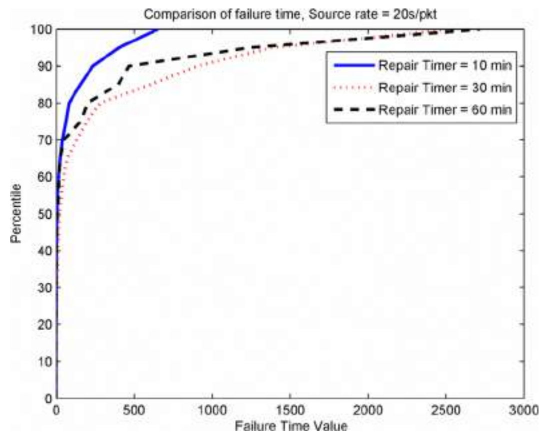

Fig. 11. CDF: Loss of connectivity for different Global Repair Period, Packet Rate 20s/pkt.

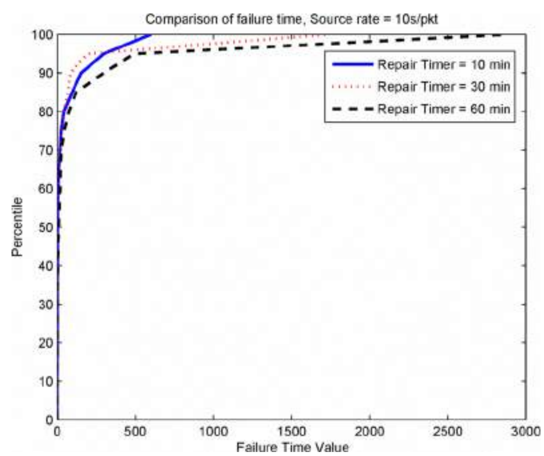

Fig. 12. CDF: Loss of connectivity for different Global Repair Period, Packet Rate 10s/pkt.

Figure 13 shows effect of DAG Global repair timer period on control traffic. As the period to emit new DAG sequence increases, the amount of control traffic decreases because the trickle interval gets larger for each node, which is pretty intuitive. However this smaller amount of control traffic comes at a price of increased time for loss of connectivity.

VI. Conclusion and Future Works

From the simulation study, it is clear that the use of trickle timer has the desired effect of keeping control overhead much less than the actual data packet count. As time advances, the number of control packet decreases with time making the network and the DAG stable. Also the time period for Global repair has significant effect on number of control packets, which may be used to upper bound the overhead for trade off with time with no connectivity. However, there are few incidents where this loss of connectivity time gets large to an order comparable with DAG sequence number period, mainly in cases where due to very low PDR, no DIS or DIO is heard for a long time. The protocol operation has to be modified for outdoor network cases like this.

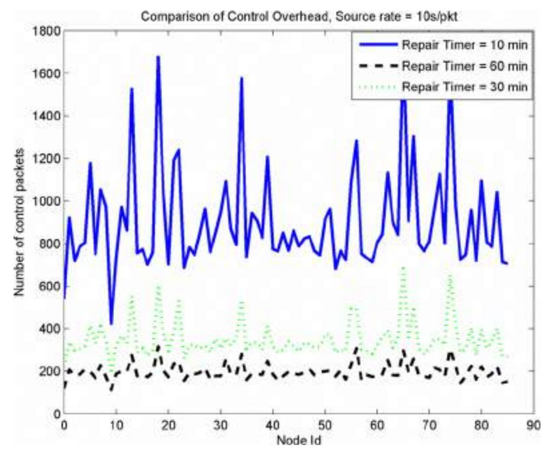

Fig. 13. Amount of control traffic for different Global Repair Timer Period.

It is necessary to implement and study the behavior of this protocol in large scale (thousands of nodes) network, which is typically the case for smart metering system. For typical smart object deployment scenarios, such as building network routing or home network routing, the traffic pattern is different (60%-70% of traffic is confined within 1-2 hop of physical distance). Path stretch for these kind of networks and very large scale networks needs to be examined and improved. We also plan to observe the effect of DAO packing (using a single DAO for all network prefixes in the sub-DAG) on amount of control overhead and stability of the system. Also, the local repair mechanism needs to be implemented, where the node does not need to wait for a new DIO and a higher DAG Sequence number to arrive in order to find a new parent. The behavior of the above need to be documented for medium and large scale network.

References

[1] "http://www.ietf.org/dyn/wg/charter/roll-charter.html."

[2] T. Winter, P. Thubert, and et al., "Rpl: Routing protocol for low power and lossy networks, draft-ietf-roll-rpl-04," November 2009.

[3] A. Brandt, J. Buron, and G. Porcu, "Home automation routing requirements in low power and lossy networks, draft-ietf-roll-home-routing-reqs-08 (work in progress)," September 2009.

[4] M. Dohler, T. Watteyne, T. Winter, and D. Barthel, "Rfc 5548,routing requirements for urban low-power and lossy networks," May 2009.

[5] J. Martocci, N. Riou, P. Mil, and W. Vermeylen, "Building automation routing requirements in low power and lossy networks, draft-ietf-roll-building-routing-reqs-07 (work in progress)," September 2009.

[6] K. Pister, P. Thubert, S. Dwars, and T. Phinney, "Industrial routing requirements in low power and lossy networks, draft-ietf-roll-indus-routing-reqs-06 (work in progress)," June 2009.

[7] J. Vasseur, M. Kim, K. Pister, and H. Chong, "Routing metrics used for path calculation in low power and lossy networks, draft-ietf-roll-routing-metrics-04 (work in progress)," December 2009.

[8] A. Varga, "The omnet++ discrete event simulation systems," in *European Simulation Multiconference (ESM'2001)*, June 2001.

[9] A. Boulis, "Castalia: Revealing pitfalls in designing distributed algorithms in wsn," in *5th international conference on Embedded networked sensor systems (SenSys'07)*, 2007.

[10] "The network simulator-2, http://www.isi.edu/nsnam/ns/."

[11] J. Jurski, "Limited ip header compression over ppp, draft-jurski-pppext-iphc-02.txt (work in progress)," March 2007.

[12] J. P. Vasseur and A. Dunkel, "Interconnecting smart objects with ip: The next internet , "http://www.thenextinternet.org"."