# A Performance Analysis of Point-to-Point Routing along a Directed Acyclic Graph in Low Power and Lossy Networks

Weigao Xie, Mukul Goyal, Hossein Hosseini, Jerald Martocci, Yusuf Bashir, Emmanuel Baccelli, Arjan Durresi

# A Performance Analysis of Point-to-Point Routing along a Directed Acyclic Graph in Low Power and Lossy Networks

W Xie*, M Goyal*, H Hosseini*, J Martocci†, Y Bashir†, E Baccelli‡ and A Durresi§
*U Wisconsin - Milwaukee, Milwaukee, WI 53201 USA, {wxie,mukul,hosseini}@uwm.edu
†Johnson Controls, Milwaukee, WI 53202 USA, {jerald.p.martocci,yusuf.bashir}@jci.com
‡INRIA, Paris, France, emmanuel.baccelli@inria.fr
§IUPUI, Indianapolis, IN 46202 USA, durresi@cs.iupui.edu

## Abstract

*The ROLL working group at IETF is currently designing RPL, the Routing Protocol for Low power and lossy networks. RPL is based on organizing the nodes in a network in one or more directed acyclic graphs (DAGs) rooted at the popular/default destinations. The DAG structure naturally supports multipoint-to-point routing towards the DAG root and point-to-multipoint routing from the root towards the nodes. The point-to-point (P2P) routing mechanism specified in RPL requires a packet to travel upwards along the DAG until it is turned downwards by a node that knows the downward route along the DAG to the packet's destination. This up and down routing along the DAG may result in significantly suboptimal P2P routes. In this paper, we quantify the difference in terms of routing cost between DAG-based P2P routes and the shortest (or minimum cost) P2P routes available in a sample network topology and make the case for a better P2P routing solution in RPL.*

## 1. Introduction

Designing a highly scalable routing protocol for large *low-power and lossy networks* (LLN), consisting of potentially thousands of memory, power and CPU constrained nodes and unreliable links, presents unique challenges. The *Routing Over Low-power and Lossy networks* (ROLL) working group in *Internet Engineering Task Force* (IETF) is currently engaged in the design of such a routing protocol, called *RPL*[1] [1], for LLN-based applications in industrial, urban, commercial building and home domains [2]–[5]. These applications have identified *multipoint-to-point* (MP2P) data collection as a dominant traffic pattern. Consequently, RPL organizes the nodes in the network as one or more *destination oriented directed acyclic graphs* (DODAGs or simply DAGs), rooted at nodes that serve as popular destinations or provide default routes to rest of the Internet [1]. The DAG structure allows a node to have multiple *parent* nodes such that forwarding a packet to any parent brings it closer to the root. The acyclic nature of the DAG is maintained by requiring a node to have a higher *rank* than any of its parents, where the DAG root maintains a rank 1 and a non-root node determines its rank by adding to the rank of its *preferred* parent a *step* value that characterizes the link between the node and the preferred parent and is calculated as specified in an *objective function* (OF) from one or more routing metrics. A node periodically advertises its DAG rank as well as the values of various routing metrics in a *DODAG Information Object* (DIO) message sent via link-local multicast to its DAG children.

While a DAG naturally supports MP2P routing towards its root, further information exchange is required to support *point-to-multipoint* (P2MP) and *point-to-point* (P2P) routing as well. The current RPL specification [1] supports the provisioning of *downwards* routes by allowing a node to advertise itself to a subset of its parents by sending them (via unicast) a *destination advertisement object* (DAO) message. A *storing* parent stores the DAOs received from its descendants and aggregates the reachability information about itself and its descendants in the DAO it sends to its parent(s). A *non-storing* parent simply adds its own address to the *reverse route stack* in the child's DAO and forwards it to its parent(s). Thus, a storing node is able to maintain *piecewise source* routes (until the next storing node) for a subset of its descendants. RPL does not impose any requirements regarding which nodes should originate a DAO and the set of parents to whom a DAO should be sent. From the reachability perspective, the best

[1] *R*outing *P*rotocol for *L*ow power and lossy networks

case scenario is when all nodes are *storing* in nature and send DAOs to all their parents. In the worst case scenario, only the DAG root will store DAOs (i.e. only the DAG root has the ability to send packets downwards) and a DAO is forwarded to only one parent (thus only one, possibly suboptimal, downwards route is available for a destination).

The DAO mechanism extends P2P routing functionality to nodes in a DAG. RPL allows a node to send a DAO to all its radio-range neighbors via link-local multicast, thereby allowing them to reach the node directly. If a destination is not in the source's radio range but the source maintains a downwards route to the destination, it can send packets to the destination along this route. Otherwise, the source sends the packet to a DAG parent. If this parent knows the downwards route to the destination, it sends the packet along that route; otherwise, the packet is sent to one of its parents. In this manner, a packet travels upwards along the DAG until it is turned downwards by a storing node that maintains a route to the packet's destination. Thus, in the best case scenario identified previously, a packet starting from the source will travel upwards along the DAG until it reaches the first common ancestor of the source and the destination. Then, the packet will travel downwards along the DAG until it reaches its destination. In the worst case scenario, where only the DAG root maintains downwards routes, the packet will have to travel all the way to the DAG root before it begins its descent towards the destination.

This *up* and *down* routing along the links in a DAG may result in a significantly worse route than the *shortest* (or the *minimum cost*) route available between a source and a destination in the network. Significant suboptimality in P2P routes is a cause of concern for many LLN applications, including those in building and home domains [4], [5], that rely on P2P communication for their operation. In this paper, we quantify the difference in terms of routing cost between DAG-based P2P routes and the shortest P2P routes available in a sample network topology and make the case for a better P2P routing solution in RPL. We also compare the routing costs for DAG-based P2P routing when *upwards to downwards* turnaround takes place at the first common ancestor of the source and destination versus the costs when the packet has to travel all the way to the DAG root before turning downwards towards its destination.

The rest of the paper is organized as follows. In the next section, we describe the characteristics of the topology used in the analysis. Section 3 presents the results of the routing cost comparison between DAG-based routes and the shortest cost routes between different source-destination pairs. Section 4 compares the routing costs for DAG-based P2P routes when the turnaround takes place at the first common ancestor between the source and the destination versus the case when the turnaround takes place at the root. Section 5 describes the difference between DAG-based and shortest cost P2P routing in terms of link-level traffic loads in the network when supporting a certain number of P2P traffic flows. Section 6 concludes the paper.

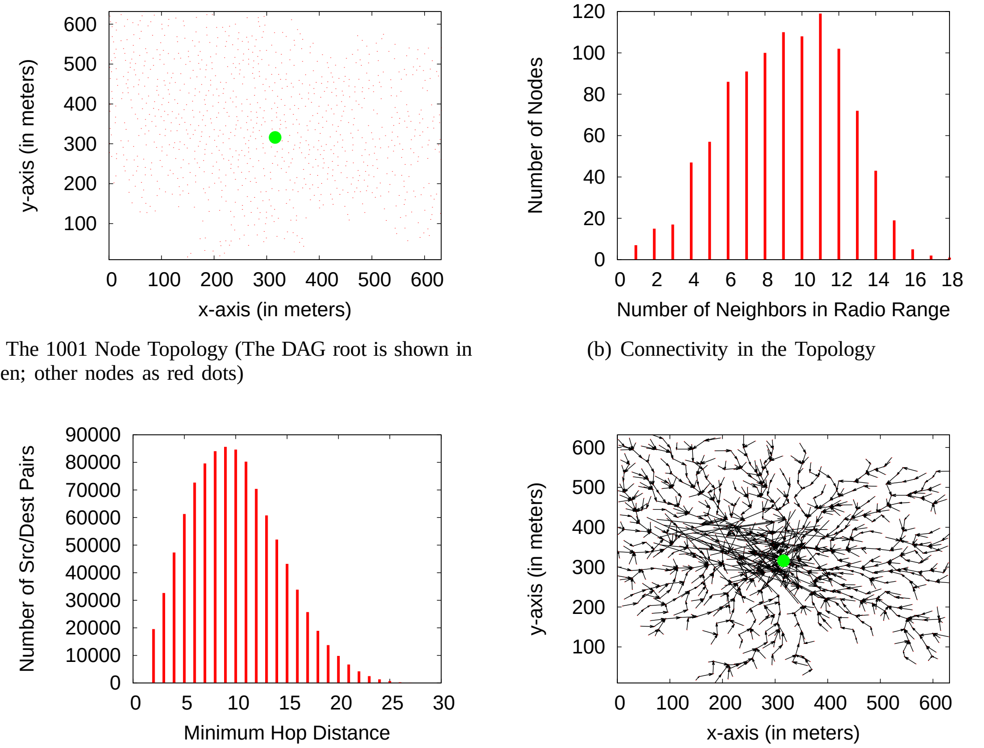
## 2. The Network Topology

The performance analysis, presented in this paper, was done on a network of 1001 nodes distributed in a $632m \times 632m$ region. This number represents the expected upper limit on the number of nodes per DAG in the future real deployments. The node locations were determined one-by-one in the following manner. The $x$ and $y$ coordinates of a new location were determined in a uniform random fashion in range $\{0m, 632m\}$ under the constraint that the minimum distance between a new location and an existing location should not be less than 10m or larger than 30m. This was done to ensure that a new location is always in the radio range of atleast one existing location. The radio range for each node in these simulations was a circle with radius 31.45m. Thus, we ensured that there were no partitions in the network topology.

Figure 1 illustrates various aspects of the network topology. Figure 1(a) gives a visual representation of the network topology and Figure 1(b) shows the connectivity in the topology in terms of the number of nodes with a given number of neighbors in their radio range. Figure 1(c) shows the number of source-destination pairs with a given *minimum* hop distance between them. For simplicity, the performance analysis presented in this paper is based on the assumption that all links have the same (or *unit*) cost. However, we repeated this analysis for more general distributions of links costs as well and found that the conclusions do not depend on the link cost function being used. Figure 1(d) shows the shortest path tree (representing a DAG), minimizing each node's cost (in this case, hop count) to reach the tree's root. In the following discussion, we refer to this tree as the DAG.

## 3. Comparing P2P Route Costs: Shortest Path Routing versus Routing Along the DAG

Figure 2 compares the costs of shortest P2P routes with the costs when P2P routing takes place along



(a) The 1001 Node Topology (The DAG root is shown in green; other nodes as red dots)

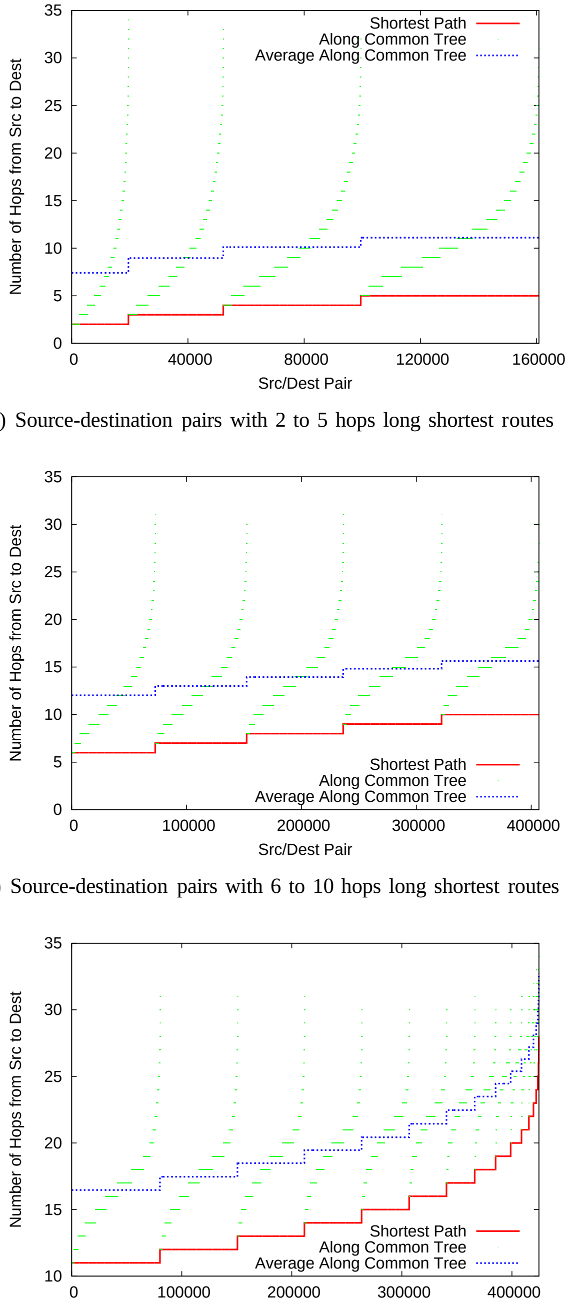
(b) Connectivity in the Topology

(c) Distribution of Minimum Hop Distance between Source-Destination Pairs

(d) The shortest path tree (DAG) based on unit link costs and thus minimizing each node's hop distance to/from the root (the DAG root is shown in green; other nodes as red dots)

Figure 1. Characteristics of the 1001 Node Topology Used in the Performance Analysis

the DAG (shown in figure 1(d) and referred to as the *common* tree in the figures) with the turnaround taking place at the first common ancestor of the source and the destination. Consider figure 2(a). The figure shows the cost comparison (when using unit link costs) for the source-destination pairs whose shortest routes are 2 to 5 hops long. Examining the cost comparison for these source-destination pairs is important because, in many LLN applications, point-to-point communication typically takes place between nodes that are located close to each other although not necessarily in each other's radio range. Since the network topology used for performance comparison consists of 1001 nodes, there are a total of 1001000 different source-destination pairs. Out of these, 160788 source-destination pairs were observed to be 2 to 5 hops apart (19538 pairs 2 hops apart; 32634 pairs 3 hops apart; 47334 pairs 4 hops apart and 61282 pairs 5 hops apart). Figure 2(a) shows the corresponding number of hops (in green) when routing between the source and destination takes place along the tree/DAG. The figure also shows (in blue) the average number of hops traversed when routing along the tree/DAG. For the nodes that are only 2, 3, 4 and 5 hops away from each other, the average number of hops when routing along the tree/DAG becomes 7.41, 8.95, 10.11 and 11.1 respectively. The corresponding worst case hop distance when routing along the tree/DAG was observed to be 34, 33, 32 and 32 respectively. As demonstrated later, this significant increase in the hop count when routing along the tree/DAG translates into significant increase in the traffic load on the links, which will result in a serious deterioration in the packet loss rate and latency for LLN applications. Figures 2(b) and 2(c) show the corresponding comparison for source-destination pairs that are 6 to 10 and higher hops away from each other respectively. These figures demonstrate that the routing along the DAG results in packets traveling much higher number of hops than what they would when using shortest routes.



(a) Source-destination pairs with 2 to 5 hops long shortest routes

(b) Source-destination pairs with 6 to 10 hops long shortest routes

(c) Source-destination pairs with shortest routes more than 10 hops long

Figure 2. Comparison of the total cost (number of hops) along the shortest (minimum hop) route from a source to a destination with the cost when routing along the tree/DAG (built to minimize each node's number of hops from the root). The results were sorted in order of increasing shortest route costs and then in order of increasing route costs along the tree/DAG.

## 4. Comparing P2P Route Costs When Routing Along the DAG: Turning Around at First Common Ancestor Versus Turning Around at Root

RPL allows a destination to advertize itself to its ancestors in the DAG via the *destination advertisement option* (DAO) mechanism. However, a node that receives routing information about a descendant in the DAG may choose not to store this information because of memory constraints. In that case, the node simply adds itself to the *reverse route stack* stored in the DAO packet and forwards it to a parent. Thus, the P2P route along the DAG from a source to a destination may not always turnaround at the first common ancestor of the source and the destination. In the worst case, a packet may have to travel all the way to the DAG's root before moving downwards towards its destination. In figure 2 discussed in the previous section, the costs associated with DAG-based routing were calculated assuming that the turnaround takes place at the first common ancestor of the source and the destination. In this section, we compare the costs associated with DAG-based P2P routing when the turnaround takes place at the first common ancestor of the source and the destination versus when the packet travels all the way to the DAG's root before turning around.

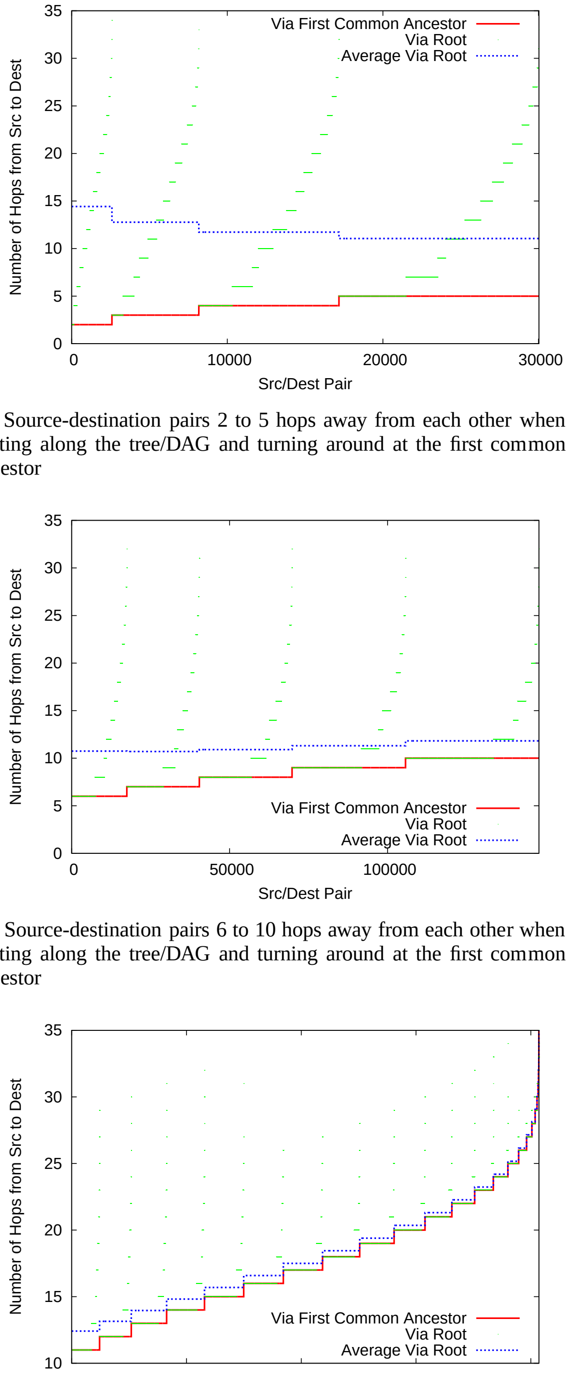
Figure 3 shows the cost comparison in two cases when all links have unit costs. Figure 3(a) shows the comparison for the source-destination pairs that are 2 to 5 hops away when routing along the DAG and turning around at the first common ancestor. Figures 3(b) and 3(c) show the comparison for the source-destination pairs that are 6 to 10 and higher hops away when routing along the DAG and turning around at the first common ancestor. For a given number of hops travelled (shown in red) along DAG when turning around at the first common ancestor of a source and a destination, these figures show the corresponding number of hops travelled (shown in green) when the packets have to go all the way to the root before turning around. These figures also show (in blue) the average hop count when the turnaround takes place at the root for the source-destination pair with a particular hop count when the turnaround takes place at the first common ancestor. As figure 3(a) shows, for the source-destination pairs that have a small hop count when the turnaround takes place at the first common ancestor, the hop count when the turnaround takes place at the root can be much higher. However, as the hop count with turnaround at the first common ancestor increases (figures 3(b) and 3(c)), the difference with the hop count with turnaround at the root decreases. As figure 3(c) shows, when a source and a destination are far apart, in most cases the root itself is the first common ancestor between the source and the destination and hence the hop count is same in both cases.

In Section 3, we demonstrated that the DAG-based P2P routing costs (with turnaround at the first common ancestor) can be significantly higher than the minimum route costs, especially when the source and the destination are close to each other (but not in the radio range). The trends described in figure 3 indicate that the DAG-based P2P routing costs further deteriorate (especially for closeby endpoints) when the turnaround takes place at the root rather than at the first common ancestor. Thus, if an LLN application employs many P2P flows and most P2P flows are between closeby endpoints, it may be beneficial to require most/all DAG nodes to store downwards routing information about their descendants.

## 5. Traffic Load on the Links: Shortest Path Routing Versus Routing Along the DAG

In this section, we demonstrate how the increase in hop-count/route-cost with DAG-based routing (with turnaround at the first common ancestor) translate in terms of increase in the traffic loads on the links. For this purpose, we chose two sets of P2P flows and calculated the link-level traffic loads on the network while carrying these flows. The first set consisted of 1000 flows selected in the following manner: each node in the network (except the root) randomly selects a node in its 2 to 5 hop neighborhood (i.e. the nodes that can be reached in 2 to 5 hops with shortest path routing) and sends 1 packet every second to this node. The second set consisted of 10000 flows with each node selecting 10 nodes in its 2 to 5 hop neighborhood and sending each one of them 1 packet per second. Then, we calculated the traffic load on each link when these flows are routed along the common tree/DAG (with turnaround at the first common ancestor) and when shortest path routing is used. The traffic load on a link is calculated simply as the sum of traffic of all the flows that pass through the link.

Figure 4(a) shows the link level traffic loads in the network with 1000 flows under unit link costs. Figure 4(b) shows the corresponding results for 10000 flows. There are a total of 9044 links in the topology and the figures do not show the links that were not used at all. The values shown in these figures were first sorted in increasing order of the traffic loads under DAG-based routing and then under shortest path routing. The traffic loads were displayed on a log scale



(a) Source-destination pairs 2 to 5 hops away from each other when routing along the tree/DAG and turning around at the first common ancestor

(b) Source-destination pairs 6 to 10 hops away from each other when routing along the tree/DAG and turning around at the first common ancestor

(c) Source-destination pairs more than 10 hops away from each other when routing along the tree/DAG and turning around at the first common ancestor

Figure 3. Comparison of the number of hops along the tree/DAG: turning around at first common ancestor (of source and destination) versus turning around at the root.All links have unit costs. The results were sorted in order of increasing costs when turning around at the first common ancestor and then in order of increasing costs when turning around at the root.

to accomodate the vast range of traffic loads. These figures clearly indicate that DAG-based routing results in large overloads on a fraction of links that end up being a part of a large number of flows. Notice that the DAG-based routing results shown in these figures were based on the turnaround taking place at the first common ancestor of the source and the destination. The traffic overloads can be expected to be much worse in the links closer to the DAG root if all the packets were to travel to the DAG root before turning around.

Figure 5 shows the topological view of the traffic loads on the links for 10000 flows under shortest path routing and DAG-based routingwith turnaround at the first common ancestor of the source and the destination. The links are color-coded in the following manner. All links with traffic load more than 100 packets/sec have been colored red. Links with traffic load between 50 and 100 packets/sec have been colored orange. Links with traffic load between 20 and 50 packets/sec have been colored green while the links with traffic load less than 20 packets/sec (but more than 0) have been colored blue. Links that are not used at all are not shown. A traffic load of 100 packets/sec or more can be considered excessive for links using popular IEEE 802.15.4 MAC/PHY protocols. The maximum data rate possible on an IEEE 802.15.4 link operating in 2.4GHz range is 250 Kbps, which translates to the maximum capacity of 208 packets/sec when the PHY-level packet size is 133 bytes (the maximum possible value). In practice, because of hardware-related issues and CSMA-based competition among nodes for packet transmission, the achievable packet rates are much smaller. A traffic load of 100 packets/sec on a link is likely to result in excessive packet loss and large delays for packets that do manage to get transmitted successfully. Even a traffic load between 50 and 100 packets/sec can be considered large and such links are likely to suffer heavy packet loss and latency.

As figure 5(a) shows, the shortest path routing was able to support 10000 flows in the sample network topology with only very few links exceeding the traffic load of 50 packets/sec. On the other hand, DAG-based routing, even when the turnaround takes place at the first common ancestor, results in a significant number of (orange/red) links having a traffic load of more than 50 packets/sec (figure 5(b)). Such heavily loaded links are likely to suffer heavy packet losses and latency. Clearly, DAG-based routing is not able to support as many P2P flows in a network as the shortest path routing.

## 6. Conclusion

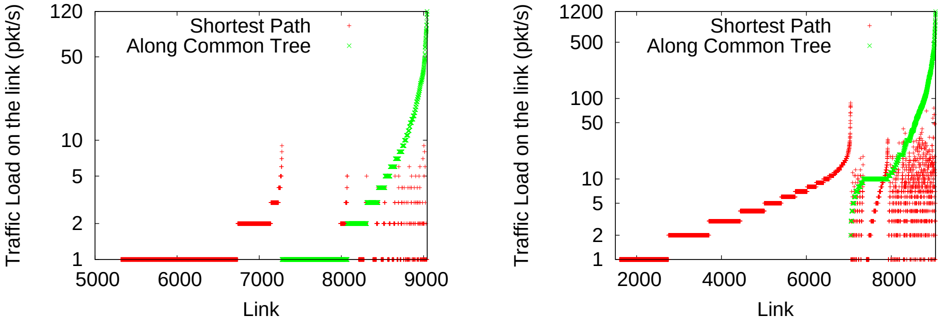
The ROLL working group in IETF is currently engaged in the design of RPL, a routing protocol for large scale low-power and lossy networks (LLNs) based on organizing the network topology along one or more *directed acyclic graphs* (DAGs) rooted at popular/default destinations. Although the *multipoint-to-point* traffic flows, naturally supported by DAGs, are common in LLN applications, the *point-to-point* routing functionality has a critical importance as well. The *point-to-point* routing solution currently available in RPL requires a packet to travel upwards along the DAG until it reaches a node that knows the downwards route to the packet's destination. In the best case scenario, the packet has to travel up the DAG until it reaches the first common ancestor of the source and the destination. In the worst case scenario, the packet has to travel up the DAG all the way to the DAG root before it can start its downwards journey towards its destination. In this paper, we demonstrated for a sample 1001 node topology that such *up and down* P2P routes can be significantly worse than the shortest cost P2P routes available in the network. The results presented in this paper clearly demonstrate the inadequacy of DAG-based P2P routing. The difference in cost between DAG-based routes and shortest cost routes is particularly appalling for the source-destination pairs that are relatively close to each other. This difference is likely to worsen as the number of nodes that constitute a DAG further increases. Clearly, there is a need to develop additional P2P routing mechanisms, either within RPL or as separate protocols, that can provide more optimal P2P routes. A scenario where the DAG-based routing is the only P2P routing option may not be acceptable to LLN applications that rely heavily on P2P flows.

## References

[1] T. Winter and P. Thubert, "RPL: IPv6 Routing Protocol for Low Power and Lossy Networks," IETF, Internet-Draft draft-ietf-roll-rpl-07, Mar. 2010.

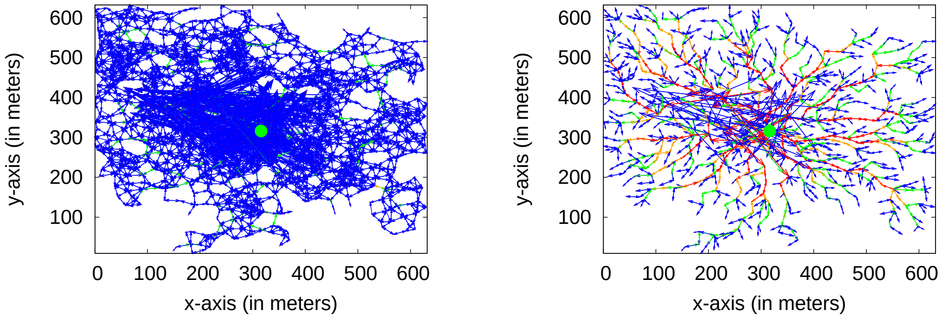
[2] K. Pister, P. Thubert, S. Dwars, and T. Phinney, "Industrial Routing Requirements in Low-Power and Lossy Networks," IETF, Request For Comments 5673, Oct. 2009.

[3] M. Dohler, T. Watteyne, T. Winter, and D. Barthel, "Routing Requirements for Urban Low-Power and Lossy Networks," IETF, Request For Comments 5548, May 2009.



(a) Unit Cost Links, 1000 Flows (b) Unit Cost Links, 10000 Flows

Figure 4. Comparison of the traffic load on the links under shortest path routing versus routing along the DAG. The results are sorted in increasing order of link loads under DAG-based routing and then in increasing order of link loads under shortest path routing.



(a) Unit Cost Links, 10000 Flows, Shortest Path Routing (b) Unit Cost Links, 10000 Flows, Routing Along DAG

Figure 5. Topological view of the link-level traffic loads for 10000 flows under shortest path routing and DAG-based routing. The links are color coded based on their traffic loads as follows: Red (traffic load more than 100 pkts/sec), Orange (between 50 and 100 pkts/sec), Green (between 20 and 50 pkts/sec), Blue (more than 0 but less than 20 pkts/sec).

[4] J. Martocci, P. DeMil, W. Vermeylen, and N. Riou, “Building Automation Routing Requirements in Low Power and Lossy Networks,” IETF, Internet-Draft draft-ietf-roll-building-routing-reqs-09, Jul. 2010.

[5] A. Brandt, J. Buron, and G. Porcu, “Home Automation Routing Requirements in Low Power and Lossy Networks,” IETF, Internet-Draft draft-ietf-roll-home-routing-reqs-11, Jan. 2010.