

A Performance Analysis of P2P Routing along a DAG in LLNs

Abstract

In this draft, we analyze the *point-to-point* (P2P) routing performance of RPL [1] by comparing the *shortest* P2P route costs in a sample network topology with the corresponding costs when routing along a tree built to minimize the routing costs to the tree's root.

1 Introduction

In this draft, we analyze the *point-to-point* (P2P) routing performance of RPL [1] by comparing the *shortest* P2P route costs in a sample network topology with the corresponding costs when routing along a tree built to minimize the routing costs to the tree's root. Such a tree represents an RPL *directed acyclic graph* (DAG), which, in general, allows a node to have multiple parents as opposed to a single parent allowed in a tree.

Point-to-point routing along a DAG, as specified in RPL specification [1], works in the following manner. Two nodes, in each other's radio range, can send packets to each other directly. However, if two nodes are not in each other's radio range, they can send packets to each other only along the DAG links, which means that a packet travels up the DAG until it reaches a node that knows of the downwards route to the destination and then it travels down the DAG towards its destination. A node that belongs to a DAG must maintain one or more parents in the DAG that serve as the default *next hops* for packet forwarding. In addition, a node may optionally maintain *downwards* routing information for its descendants in the DAG. The DAG root may need to maintain downwards routing information for all the nodes in the DAG. In the best case *point-to-point* scenario in a DAG, the packet travels up the DAG only till it encounters the first common ancestor of the source and the destination. In the worst case, the packet may need to travel all the way to the DAG's root before it starts its downward journey towards the destination. The routing cost along a DAG for a source-destination pair refers to the sum of the link costs along the route between the source and the destination along the DAG (up the DAG and then down). While routing along a DAG is constrained to the links in the DAG, the *shortest* route from a source to a destination is calculated without any restrictions.

2 The Network Topology

The performance analysis, presented in this draft, 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

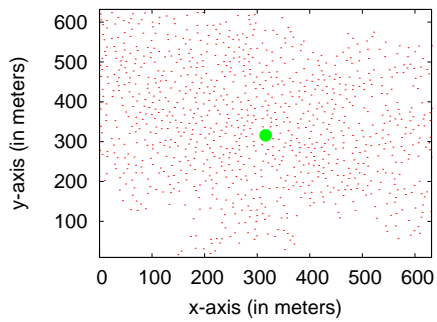
per DAG in the future real deployments. The node locations were determined one-by-one in the following manner. The x and y coordinates of a new location were determined in a uniform random fashion in range $\{0m, 632m\}$ under the constraint that the minimum distance between a new location and an existing 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. The performance analysis presented in this draft is based on both *unit* link costs as well as link costs distributed between 1 and 16 in the following manner. As per RPL specification [1], the link cost values 1, 4 and 16 signify a perfect, normal and an almost unusable link respectively. The link cost between two neighbor nodes in the network topology was determined as 2^x rounded to the closest integer, where x is a real number uniformly distributed over range $[0, 4]$. The same cost value was used for both directions of the link. Figure 1(d) shows the distribution of link costs calculated in this manner. Figures 1(e) and 1(f) show the shortest path tree (representing a DAG), minimizing each node's cost to reach the tree's root, based on the unit link costs and the link costs calculated in the manner described above respectively.

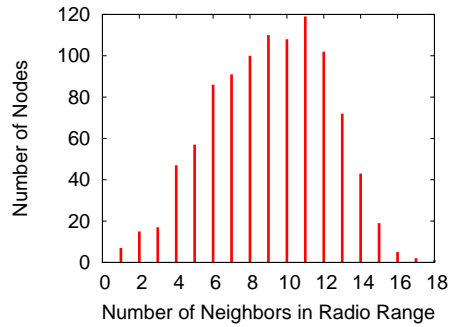
3 Comparing P2P Route Costs: Shortest Path Routing versus Routing Along the Tree/DAG

Figures 2 and 3 compare the costs of shortest routes with the costs when routing along a DAG. In these figures, the DAG is represented by a shortest path tree, referred to as the *common* tree in the figures, minimizing each node's cost to reach the tree's root. Figure 2 shows the results when using unit link costs (i.e. the shortest routes are the minimum hop routes) and Figure 3 shows the results when link costs are distributed between 1 and 16 in the manner described earlier.

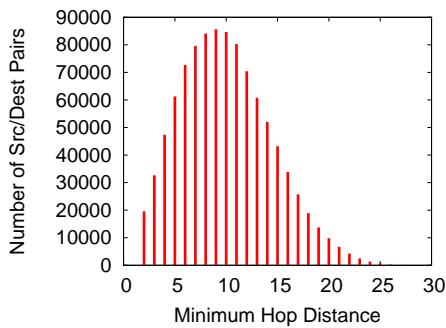
Consider figure 2(a). The figure shows the cost comparison (when using unit link costs) for the source-destination pairs whose shortest routes are 2 to 5 hops long. Examining the cost comparison for these source-destination pairs is important because, in many LLN applications, point-to-point communication 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



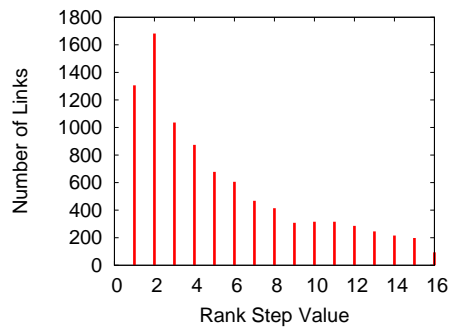
(a) The 1000 Node Topology (The DAG root is shown in green; other nodes as red dots)



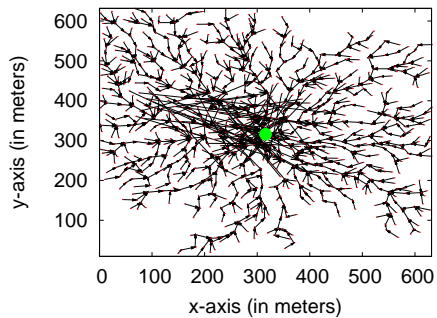
(b) Connectivity in the Topology



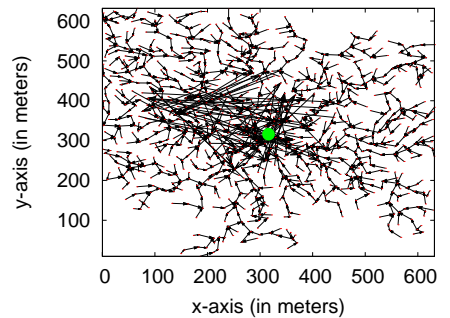
(c) Distribution of Minimum Hop Distance between Source-Destination Pairs



(d) Distribution of Link Costs



(e) The shortest path tree (DAG) based on unit distance to/from the root (the tree/DAG root is shown in green; other nodes as red dots)



(f) The shortest path tree (DAG) minimizing each node's routing cost to/from the root. Based on the link costs between 1 and 16 as shown in Fig 1(d). The tree/DAG root is shown in green; other nodes as red dots.

Figure 1: Characteristics of the 1000 Node Topology Used in the Performance Analysis

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.

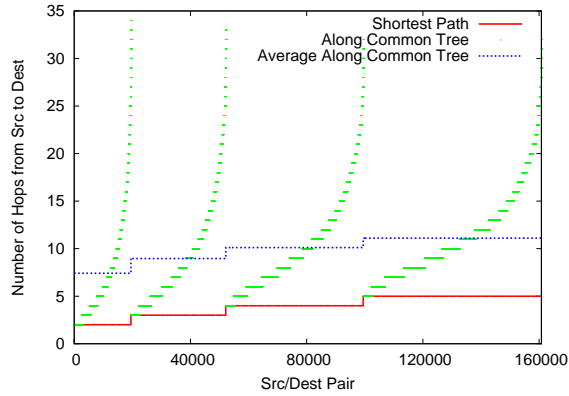
Figure 3 shows the cost comparison between shortest routes and routes along a DAG when link costs are distributed between 1 and 16. Figure 3(a) shows the comparison for source-destination pairs that have shortest route costs less than 10. Figures 3(b) and 3(c) show the comparison for source-destination pairs that have shortest route costs from 10 to 30 and higher respectively. These results are similar to the results obtained with unit link costs and confirm that the routing along the DAG may result in much higher route costs than shortest (or minimum cost) routes.

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

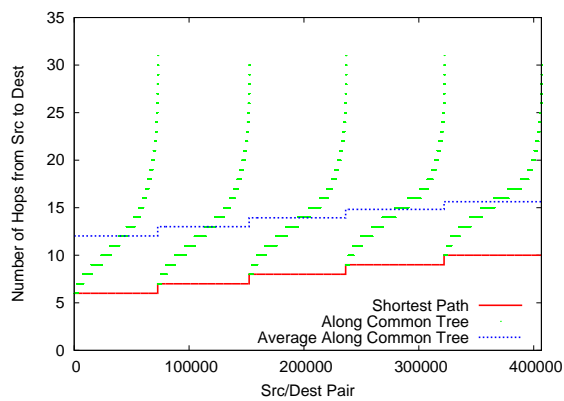
RPL allows a destination to advertize itself to its ancestors in the DAG via the *destination advertisement option* (DAO) mechanism. However, a node that receives routing information about a descendant in the DAG may choose not to store this information because of memory constraints. In that case, the 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 figures 2 and 3 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 4 shows the cost comparison in two cases when all links have unit costs. Figure 4(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 4(b) and 4(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 4(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 4(b) and 4(c)), the difference with the hop count with turnaround at the root decreases. As figure 4(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.

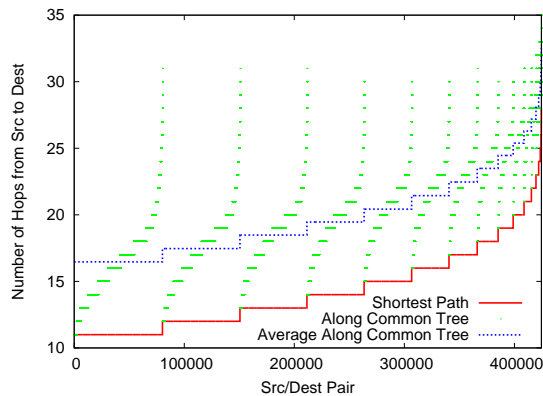
Figure 5 shows the cost comparison when link costs are distributed between 1 and 16. These results



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

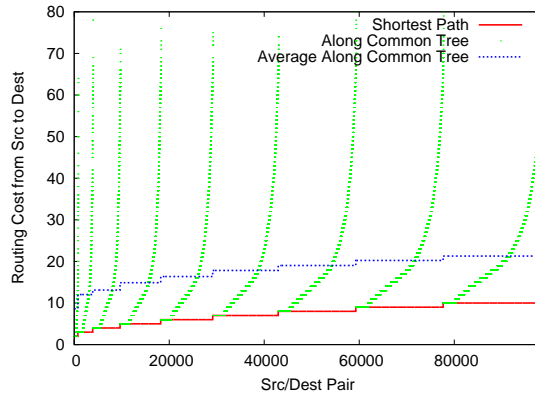


(b) Source-destination pairs with 6 to 10 hops long shortest routes

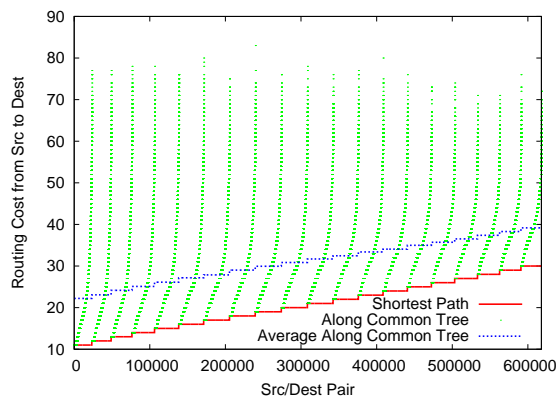


(c) Source-destination pairs with shortest routes more than 10 hops long

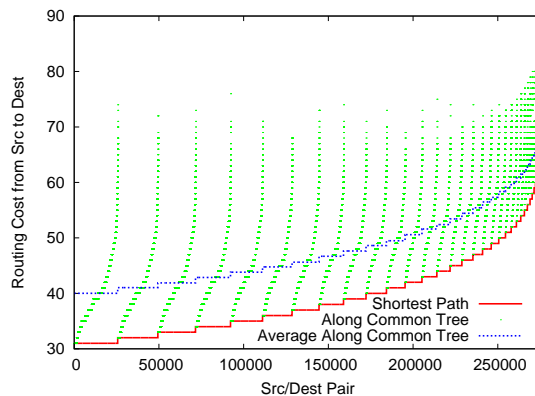
Figure 2: Comparison of the total cost (number of hops) along the shortest (minimum hop) route from a source to a destination with the cost when routing along the tree/DAG (built to minimize each node's number of hops from the root). The results were sorted in order of increasing shortest route costs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



(a) Source-destination pairs with shortest path costs less than 10

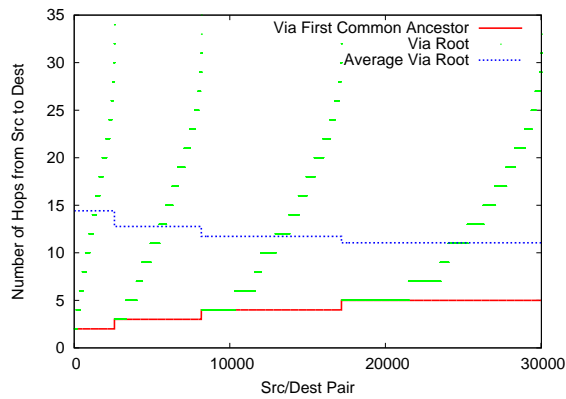


(b) Source-destination pairs with shortest path costs between 11 and 30

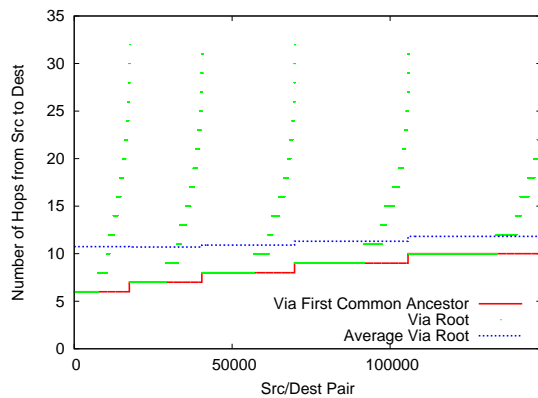


(c) Source-destination pairs with shortest path costs higher than 30

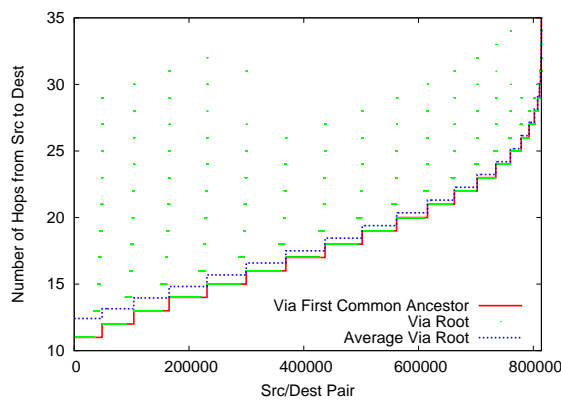
Figure 3: Comparison of the total cost along the shortest route from a source to a destination with the cost when routing along the tree/DAG (built to minimize each node's routing cost to/from the root). The link level costs were distributed between 1 and 16. The results were sorted in order of increasing shortest route costs and then in order of increasing route costs along the tree/DAG.



(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 4: 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.

follow the same trend as the results with unit link costs. Routing cost when the turnaround takes place at the root could be significantly higher if the cost is small when turnaround takes place at the first common ancestor (Figure 5(a)). As the routing cost with turnaround at the first common ancestor increases (Figure 5(b)), the difference with turnaround at the root decreases. When routing cost with turnaround at the first common ancestor itself is significant (Figure 5(c)), generally the root itself is the first common ancestor and hence the routing cost is same in both cases.

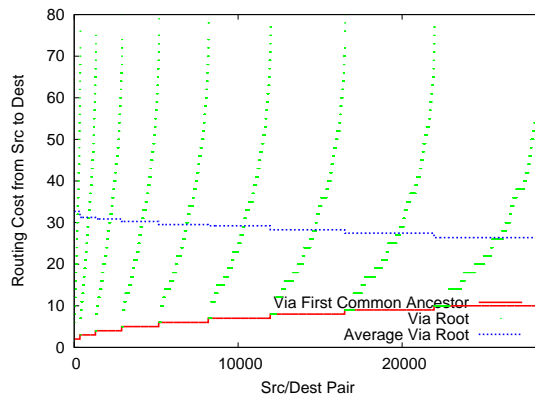
In Section 3, we demonstrated that the DAG-based P2P routing costs (with turnaround at the first common ancestor) can be significantly higher than the minimum route costs, especially when the source and the destination are close to each other (but not in the radio range). The trends described in figures 4 and 5 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 Tree/DAG

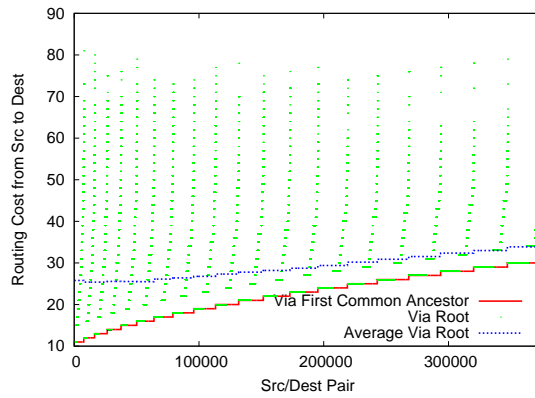
In this section, we demonstrate how the increase in hop-count/route-cost with DAG-based routing (with turnaround at the first common ancestor) translate in terms of increase in the traffic loads on the links. For this purpose, we chose two sets of P2P flows and calculated the link-level traffic loads 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.

Figures 6(a) and 6(b) show the link level traffic loads in the network with 1000 flows under unit link costs and costs distributed between 1 and 16 respectively. Figures 6(c) and 6(d) show 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 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.

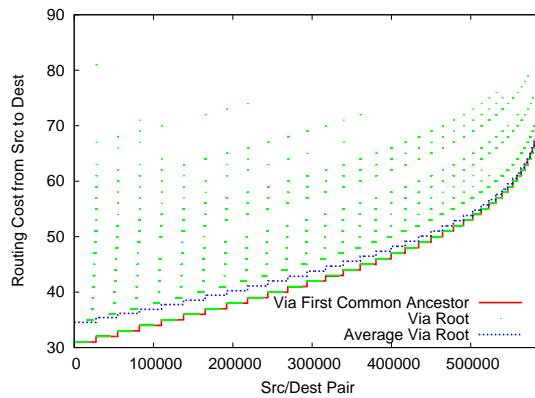
Figure 7 shows the topological view of the traffic loads on the links for 10000 flows under shortest path routing and DAG-based routing with turnaround at the first common ancestor of the source and the destination. The links are color-coded in the following manner. All links with traffic load more than 100 packets/sec have been colored red. Links with traffic load between 50 and 100 packets/sec



(a)



(b)



(c)

Figure 5: Comparison of the routing cost along the DAG: turning around at first common ancestor (of source and destination) versus always traveling via the root.

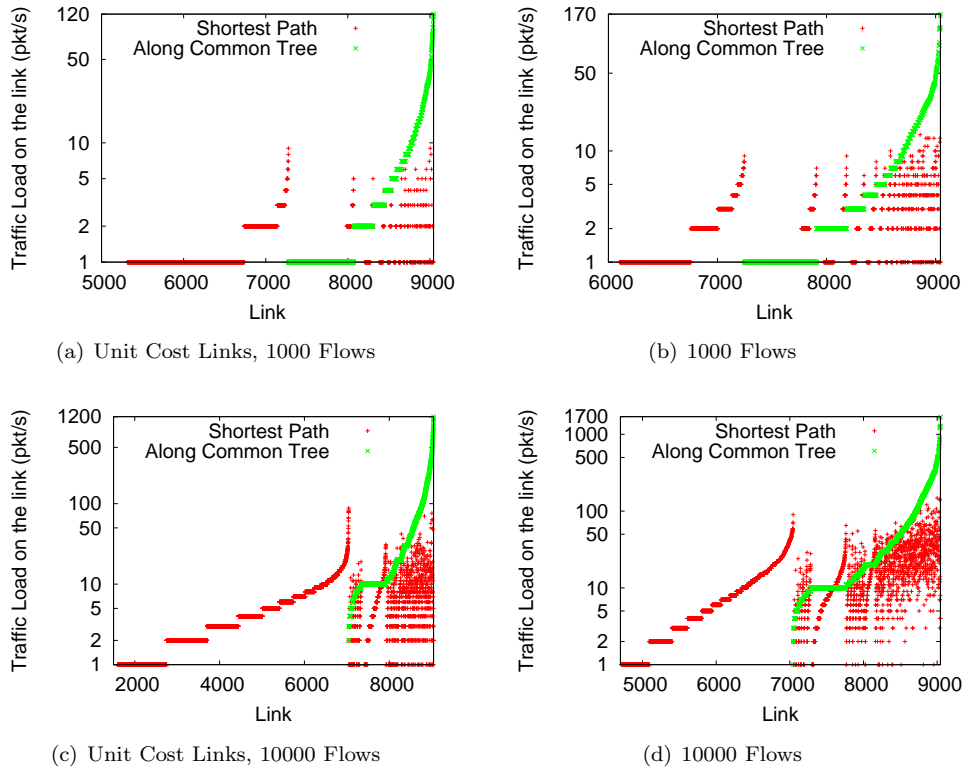
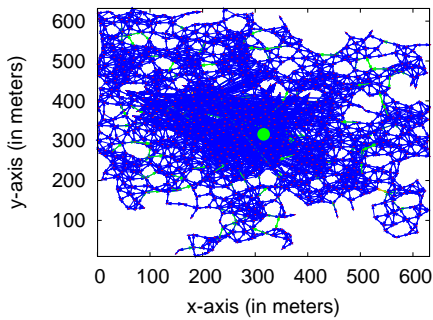
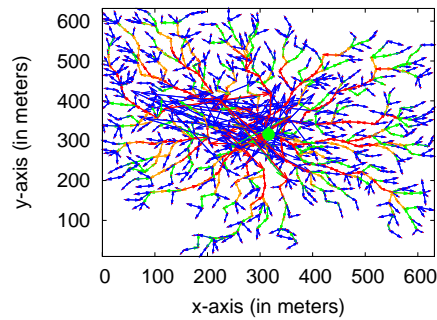


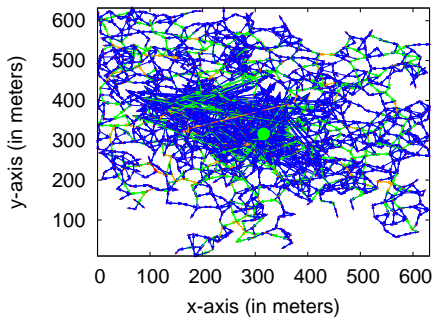
Figure 6: Comparison of the traffic load on the links under shortest path routing versus routing along the DAG. The results are sorted in increasing order of link loads under DAG-based routing and then in increasing order of link loads under shortest path routing.



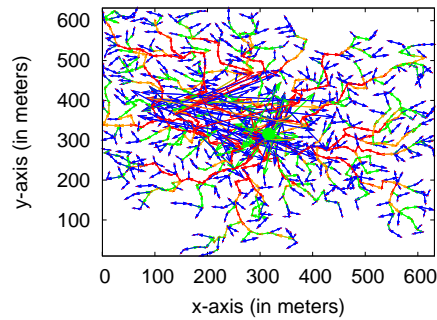
(a) Unit Cost Links, 10000 Flows, Shortest Path Routing



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



(c) Link Costs Distributed between 1 and 16, 10000 Flows, Shortest Path Routing



(d) Link Costs Distributed between 1 and 16, 10000 Flows, Routing Along DAG

Figure 7: 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).

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 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 figures 7(a) and 7(c) show, 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 (figures 7(b) and 7(d)). 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

In this draft, we compared the P2P routing cost of DAG-based routing with optimal shortest path routing for a sample 1000 node topology. The results clearly demonstrate the inadequacy of DAG-based P2P routing. The difference in cost between DAG-based routing and shortest path routing is particularly appalling for the source-destination pairs that are 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. Internet-Draft draft-ietf-roll-rpl-06, IETF, February 2010.