Lab 2 Report

Anthony Weems

October 28, 2015

1 Proofs

1.1 GPSR is non-optimal

We will show that GPSR is non-optimal by counter example. Assume that we have found a path using GPSR. Such a path must have a first node, that was closest to the sink. It is possible to construct a graph such that the first node leads the GPSR algorithm along a sub-optimal path because of the initial decision to follow the closest node.

1.2 Dijkstra Min-Hops is better, but non-optimal

Using Dijkstra's algorithm without regard for weight between nodes will provide a result similar to GPSR, but completely disregard all latency requirements. It is simple to construct a graph containing an obvious shortest path (in terms of hops), but terribly in-efficient with regard to latency. We can construct such a graph by initially setting all weights to 1, running Dijkstra's algorithm (min-hop), and then modifying the weights of all nodes in the path to have an extremely high latency. Such a graph will return the same path with a min-hop metric, but return a much more efficient path using a min-latency metric.

2 Efficiency Analysis

2.1 Memory Efficiency of Graph

The data structure used for this lab consists of a map of vertices to a map of verticies to edges. In essence, each vertex holds a map with keys for each neighbor within the transmission range. The space complexity of this representation is $O(V * E)$.

2.2 Runtime Complexity of Graph

Creating the data structure is rather fast as it only requires examining each edge in the graph. The runtime complexity of creation is $O(E)$.

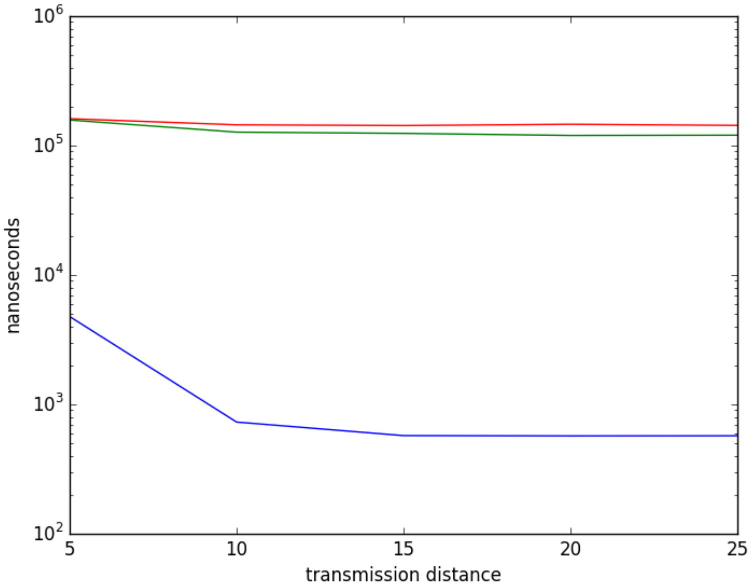
2.3 Runtime Complexity of Dijkstra

The data structure used within Dijkstra's algorithm allows $O(1)$ lookup of neighbors by allowing us to search for an edge between to vertices. Therefore, the overall runtime complexity of Dijkstra's algorithm is $O(V^2)$. Worst-case, the algorithm will examine the metric between every pair of nodes because we do not use a priority queue in our algorithm.

3 Runtime Efficiency

The graph below displays the runtime of our three algorithms:

GPSR (blue), Dijkstra Min-Latency (green), and Dijkstra Min-Hops (red).



The success rate of all algorithms was 100% for all tested transmission ranges in the large input testcase.