

Computation of Minimal Uniform Transmission Power in Ad Hoc Wireless Networks *

Qing Dai and Jie Wu
Department of Computer Science and Engineering
Florida Atlantic University
Boca Raton, FL 33431
{qdai,jie}@cse.fau.edu

Abstract

Power conservation is a critical issue for ad hoc wireless networks. The main objective of the paper is to find the minimum uniform transmission power of an ad hoc wireless network, where each node uses the same transmission power, while maintaining network connectivity. Three different algorithms, binary search, Prim's MST and its extension are developed to solve the problem, and their performance is compared by simulation study together with Kruskal's minimum spanning tree (MST), a known solution proposed by Ramanathan and Rosales-Hain for topology control by transmission power adjustment. Our results show that Prim's MST outperforms both Kruskal's MST and binary search. The performance between Prim's MST implemented with binary heap and Fibonacci heap is fairly close.

Keywords: *ad hoc wireless network, graph connectivity, minimum spanning tree, power control, transmission power*

1 Introduction

An ad hoc wireless network is an infrastructureless network. The end nodes establish connections by themselves without a base station, and communicate with each other in a multi-hop manner. Each node, typically a mobile computing device, is powered by battery. The limit of battery life places a constraint on the power consumption. It is desirable for routing algorithms to select route and transmission power that optimize energy efficiency when possible. In the routing of the ad hoc wireless network, it has been proposed that the transmission power required to support a link between two nodes is

$$P_{ij} = r^\alpha$$

*The work was supported in part by a grant from Motorola Inc., and NSF grant CCR 9900646, and grant ANI 10073737.

where r is the distance between node i and node j , and α is between 2 and 4. The transmission power cannot be reduced without limit, since the transmission range of a node will be shortened along with the reduction of transmission power. When the transmission power is too low, the network could suffer from partition.

The topology of an ad hoc network depends on several uncontrollable factors such as host mobility, interference, and on several controllable ones such as antenna orientation and transmission power. This paper deals with adjusting the controllable parameters to create desired topology to meet certain criteria. Specifically, our objective is to minimize a uniform transmission power while maintaining the network connectivity at the same time.

There are different alternatives in choosing the metrics for transmission power. Both the power over single nodes and the total power over all nodes could be used [3, 4, 5, 6]. In this paper, power over single node is used, because we believe it is more practical. If, to obtain a lower overall power, certain nodes must transmit at an extra high power level, this situation might not be acceptable in some cases. Moreover, it is assumed that each node uses the identical transmission power and therefore reaches the same transmission range. The unit disk graph $G(V, E)$ is typically used to model ad hoc wireless networks under this situation, where two nodes are connected when their distance is within the transmission range. Throughout the paper, we consider ad hoc wireless networks in which the node locations are fixed, or are snapshot of the network at a particular time frame. The problem of focus is to develop algorithms that find the minimum uniform transmission range that keeps the network connected, assuming global information of node location, link state information, and fixed network topology.

To find this minimum uniform transmission range, we propose three different algorithms, binary search, Prim's minimum spanning tree (MST), and its extension with Fi-

MinRange-Kruskal

```
1  $minRange \leftarrow 0$ 
2 sort all edges in a non-decreasing order
3 initialize  $n$  clusters, one per node
4 while  $numberOfClusters > 1$ 
5   for each  $(u, v)$  in the sorted order
6     do if  $cluster(u) \neq cluster(v)$ 
7       merge  $cluster(u)$  with  $cluster(v)$ 
8        $minRange \leftarrow distance(u, v)$ 
9 return  $minRange$ 
```

Figure 1. Algorithm I: compute $minRange$ by Kruskal's MST.

MinRange-BinarySearch

```
1  $min \leftarrow 0, max \leftarrow \sqrt{2} \ell$ 
2 while  $min < max - 1$ 
3    $curRange \leftarrow \lfloor (min + max) / 2 \rfloor$ 
4    $G \leftarrow generateUnitDiskGraph(curRange)$ 
5   if  $G$  is connected
6      $max \leftarrow curRange$ 
7   else
8      $min \leftarrow curRange$ 
9 return  $minRange \leftarrow max$ 
```

Figure 2. Algorithm II: compute $minRange$ by binary search.

bonacci heap implementation. The performance of the three algorithm as well as a well-known solution based on Kruskal's MST by Ramanathan and Rosales-Hain [3] are evaluated by computing the minimum uniform transmission range, $minRange$, in simulation study. A network of n nodes, which are randomly distributed over a region of size $\ell \times \ell$, is generated. Each node has multiple transceivers, and can thus support any multicast sessions within its transmission range. A graph is constructed according to the node location and the transmission range information; that is, an edge between nodes u and v is added to the graph if u and v are within the transmission range of each other. The performance of different algorithms to find the $minRange$ is compared. Our simulation results show that Prim's MST and its extension outperform Kruskal's MST and binary search under the given conditions.

This paper is organized as follows: Section 2 proposes three algorithms, Algorithm II, III and IV. Kruskal's MST by Ramanathan and Rosales-Hain is also reviewed (Algorithm I). Section 3 shows simulation results. Section 4 concludes this paper and discusses some future work.

2 Algorithms

Three algorithms are proposed to solve the problem: Algorithm II uses the binary search technique, Algorithms III and IV are Prim's MST with either binary heap or Fibonacci heap implementation. Ramanathan and Rosales-Hain's algorithm based on Kruskal's MSG is also reviewed, presented as Algorithm I.

The notations used in this section are as follows: in an area of size $\ell \times \ell$, the number of nodes is noted as n . D is the largest possible distance between any two nodes in this area, or Diameter, which is $\sqrt{2} \ell$. In the undirected graph representation of the area, V represents the number of vertices, which is equal to n , and E is the number of edges. An edge connecting nodes u and v is noted as (u, v) .

2.1 Algorithm I: Kruskal's Minimum Spanning Tree (MST)

Algorithm I is proposed by Ramanathan and Rosales-Hain [3]. An MST was constructed using Kruskal's algorithm [1]. Briefly, each node is initialized as a separate connected component. Edges are sorted first, and then traversed in a non-decreasing order. An edge is added to the MST whenever it connects two connected components, until all nodes are included in a single connected component. The last edge added to MST, which is the largest edge in Kruskal's MST, will be our minimum uniform range of the network (see Figure 1). The proof of correctness was provided in the same paper [3].

The construction of MST takes $O(E \lg E)$, which will be $O(V^2 \lg V^2)$ for a complete graph. The traversal of MST takes only linear time $O(V)$ before all nodes belong to a single connected component. Therefore, the overall asymptotical complexity of Algorithm I is $O(V^2 \lg V)$.

In this algorithm, all edges in the graph are sorted first, which costs $O(E \lg E)$. In reality, the transmission range can be determined without going through all the edges (in the sorted order). Therefore, some efforts in the sorting process could be wasted.

2.2 Algorithm II: Binary Search

Algorithm II uses the brute force approach. Ranging from 0 to the longest possible transmission range, the diameter D of the area, we use binary search to find the lowest transmission range that keeps the network connected. For each $curRange$ tested, a unit disk graph is generated, where an edge (u, v) is added to the graph if the distance between u and v is less than $curRange$. The value of $curRange$ is then either decreased or increased depending on whether the resultant unit disk graph is connected or not (see Figure 2).

In this algorithm, it is important to decide when the

MinRange-Prim

```

1  unreachedNodes ←  $V[G]$ 
2  for each  $u \in unreachedNodes$ 
3    do  $key[u] \leftarrow \infty$ 
4   $key[root] \leftarrow 0$ 
5   $parent[root] \leftarrow NIL$ 
6  while  $unreachedNodes \neq \emptyset$ 
7    do  $u \leftarrow extractMin(unreachedNodes)$ 
8      for each  $v \in Ad[u]$ 
9        do if  $v \in unreachedNodes$  and
               $distance(u, v) < key[v]$ 
10         then  $parent[v] \leftarrow u$ 
11               $key[v] \leftarrow distance(u, v)$ 
12  traverse the MST, find the longest edge  $minRange$ 
13  return  $minRange$ .
```

Figure 3. Algorithm III: Prim's Minimum Spanning Tree.

binary search is completed, because theoretically, the *curRange* adjustment could continue forever. We decide to use integer distance, and terminate the searching process whenever the upper and lower boundary differs by less than 1 ($max - min \leq 1$). This generally gives good performance in practice. Though the *minRange* obtained should be fairly close, it might not be the 'ultimate' minimum transmission range value, since the computation depends on the granularity of D . Note that, the precise *minRange* can be obtained with a slight modification. We can sort all edges by distance first, then perform binary search on the array of edge weights. This modification achieves precision at the cost of higher complexity, since the cost of sorting is $O(E \lg E)$, and is not adopted in this paper.

The complexity of this approach can be calculated as follows: each connectivity checking costs $O(V + E)$, with either depth-first search or breadth-first search. The overall complexity of the algorithm is $O((V + E) \lg D)$, or $O(V^2 \lg D)$ at the most. $\lg D$ indicates how many times we need to perform connectivity check, which depends on the area size. For an area of fixed size, $\lg D$ can be treated as a constant.

2.3 Algorithms III and IV: Prim's MST

Algorithms III and IV are modifications of Algorithm I, both involving MST construction. In both Algorithms III and IV, a modified Prim's algorithm [1] is used in building the MST (see Figure 3). Prim's MST starts with an arbitrary root and constructs a single tree, until it spans all the vertices. At each step, an edge of smallest possible distance that reaches a non-tree node is added, and at any stage, all

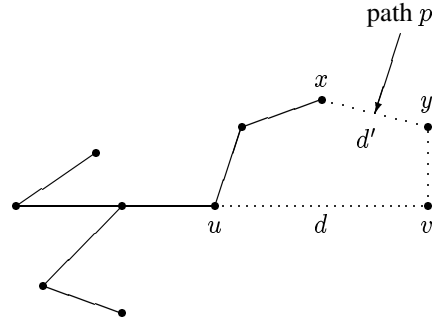


Figure 4. Proof of Theorem 1.

nodes in Prim's MST forms a single tree. To facilitate the MST construction process, a *key* for each node is maintained to represent its distance to the tree, and whenever a new node is included, the *key* value of its direct non-tree neighbors are updated. After the MST is constructed, the tree is traversed and the maximum edge is the minimum uniform range (see Theorem 1 and its proof).

It is interesting to note that the traditional minimum spanning tree algorithm can be applied to different problems. The original MST algorithms minimize the total weight of the tree. In our problem, the objective is to minimize a single uniform transmission range that keeps the network connected. In another word, it is the largest edge in the spanning tree that is minimized. The following theorem shows that the largest edge selected in Prim's MST algorithm is the minimum uniform transmission range.

Theorem 1 *The longest edge in Prim's MST is the minimum uniform transmission range.*

Proof: Suppose the longest edge in Prim's MST is (u, v) with $d = distance(u, v)$ (see Figure 4). And suppose to the contrary that d is not minimum, then there must exist a path p to connect u and v , and on this path, each of the edges is less than d . By the time (u, v) is being added to change the MST from T_i to $T_{i+1} = T_i \cup \{(u, v)\}$, assume vertex u is in the MST, and v is not, without loss of generality. If v is chosen to be included to the MST, then it must have the smallest key among all vertices currently not in the tree. Meanwhile, if p exists, then there must exist an edge (x, y) with $d' = distance(x, y)$ on path p , where x is in the tree, and y is not. (In extreme cases, x could be u , or y could be v , but not both). d' is less than d , because on path p , each edge is less than d . Therefore, y should have a smaller key than v . According to Prim's algorithm, y should be added to the MST instead of v . This is contrary to the result of Prim's algorithm, where v is added. Therefore, this path p , in which every edge is smaller than d , does not exist, and d is the minimum. \square

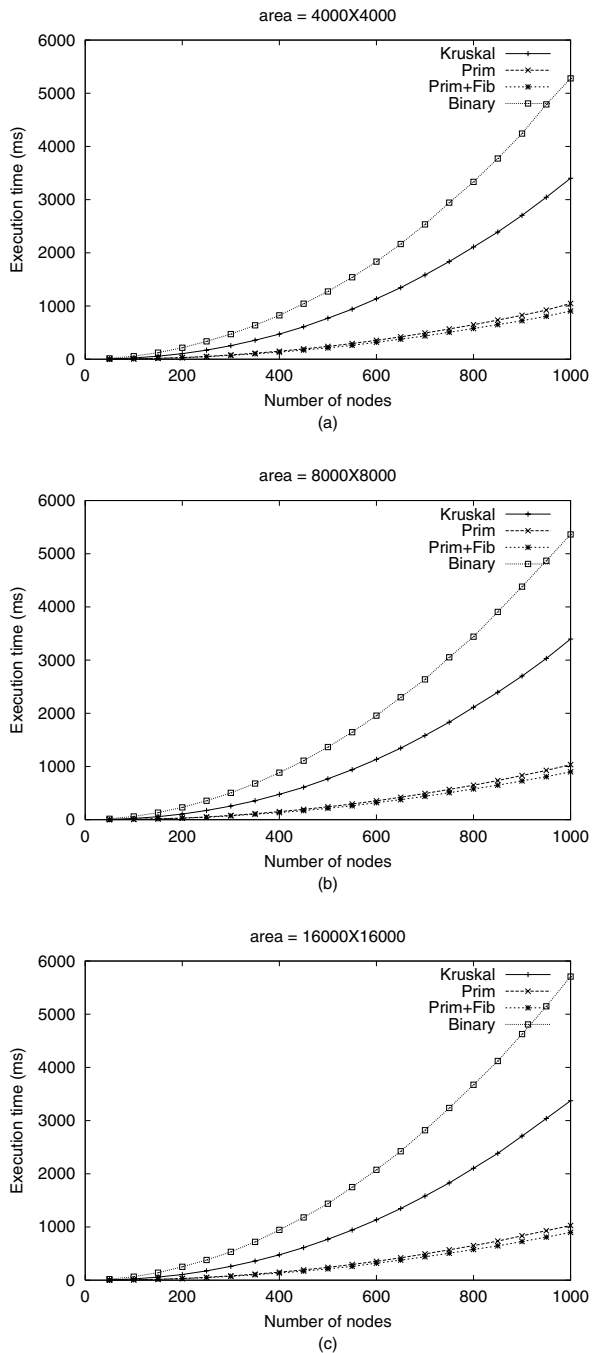


Figure 5. Simulation results. Number of nodes from 25 to 1000, in an area of 4000 × 4000 (a), 8000 × 8000 (b), 16000 × 16000 (c).

Algorithm III uses a binary-heap implementation, with a complexity of $O(E \lg V)$, which will be $O(V^2 \lg V)$ in worst case, and is asymptotically the same to that of Algorithm II. Because Prim's algorithm has a Fibonacci heap implementation, which can speed up to run in $O(E + V \lg V)$, or $O(V^2)$ for a complete graph [1], Algorithm IV takes advantage of this property. It is the same algorithm as Algorithm III, but uses the Fibonacci heap implementation. Fibonacci heap is a data structure based on binomial heaps. It is more relaxed so that work to maintain the structure can be delayed until it is convenient to perform, therefore allowing for improved asymptotic bounds[1]. Both Algorithms III and IV are implemented and their performance is compared through simulation study.

3 Simulation and Discussion

A network of n nodes, which are randomly distributed over an area of size $\ell \times \ell$, is generated. We compute the minimum transmission range using all four different algorithms described in the previous section, and record their execution time. The first set of simulation uses the same node numbers, from $n = 50$ to 1000, but in different sizes of area, $\ell = 4000, 8000$ and 16000, respectively.

Our results show that both Kruskal's and Prim's MST outperform the binary search (see Figure 5). Prim's MST has a better performance compared to Kruskal's MST. Between the two Prim's implementations using binary-heap and Fibonacci heap, there is no significant difference. It might be possible that the node number is not large enough to show the difference between them. Similar results are observed with the same node numbers in an area of 8000×8000 and 16000×16000 .

To further study the effect of node number on the performance, simulation with larger numbers of nodes, up to 3300, is performed (see Figure 6). In this simulation, Prim's MST still outperforms both Kruskal's MST and binary search. The Fibonacci-heap implementation of Prim's MST does seem to outperform the binary-heap implementation, although they are fairly close to each other.

In the future, it would be interesting to see whether binary search could outperform Kruskal's MST (Algorithm I), or even Prim's MST with binary heap implementation (Algorithm III). Theoretically, the complexity of binary search is $O(V + E) \lg D$, which is close to $O(V^2)$ if D is considered constant, in contrast to the complexity of Kruskal's MST, which will be $O(V^2 \lg V)$ in the worst case. When V is small and D is large, the effect of $\lg D$ is greater than $\lg V$. But when n is extremely large and D is relatively small, binary search should eventually outperform Kruskal's MST. Although at this point, the n might be too large to have any practical value.

The above simulation studies provide us with the basic

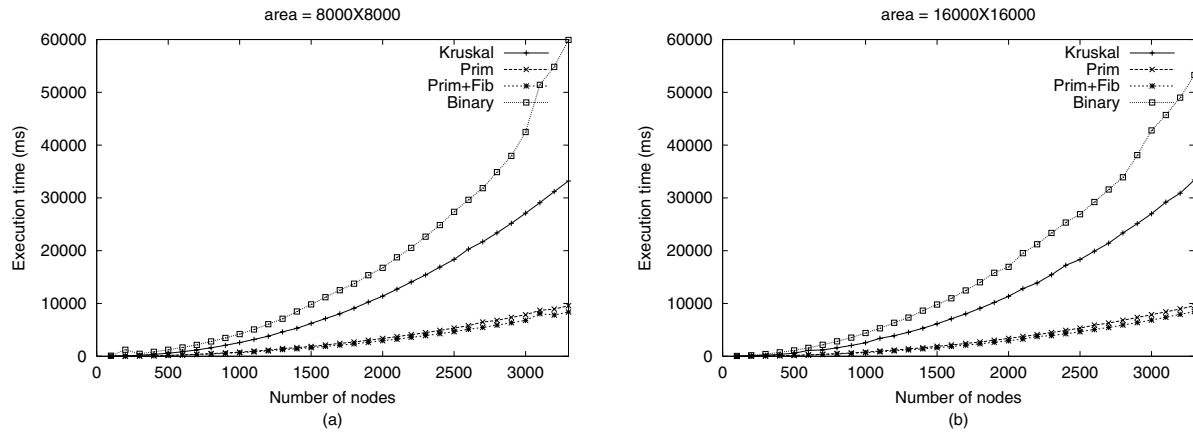


Figure 6. Simulation results. Number of nodes from 100 to 3300, distributed in 8000×8000 (a) or 16000×16000 area (b).

understandings of the relative performance between different algorithms to find the minimum uniform transmission range in ad hoc wireless networks. Their practical performance versus theoretical complexity also helps us in making decisions such as which algorithm to choose under a specific network situation with certain node number and area size.

4 Conclusion

In this paper, minimum spanning tree algorithms from classic graph theory have been applied to solve energy efficient routing problem in ad hoc wireless networks. From our simulation result, it seems that for node number up to 3000, modified Prim's algorithm has better performance than both binary search and Kruskal's MST. The performance between the two Prim's implementations is fairly close, with the Fibonacci heap implementation slightly outperforming the binary-heap implementation.

In our future study, first we plan to increase the number of nodes further, and to examine the performance of different algorithms. Meanwhile, mobility and directional antenna will also be taken into consideration in the future model.

References

- [1] T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to Algorithms*, McGraw-Hill, 1990.
- [2] J. Gomez, A. T. Campbell, M. Naghshineh, and C. Bisdikian, *Power-aware Routing in Wireless Packet Networks*, Proceedings of Sixth IEEE International

Workshop on Mobile Multimedia Communications, p380, 1999.

- [3] R. Ramanathan, and R. Rosales-Hain, *Topology Control of Multihop Wireless Networks Using Transmit Power Adjustment*, Proceedings of International Conference on Computer Communications, p404-413, Mar 2000.
- [4] S. Singh, M. Woo, and C. S. Raghavendra, *Power-aware Routing in Mobile Ad Hoc Networks*, Proceedings of MobiCom'98. Oct., 1998.
- [5] C. K. Toh, *Maximum Battery Life Routing to Support Ubiquitous Mobile Computing in Wireless Ad Hoc Networks*, IEEE Communications Magazine, 39, 6, p138-147, June 2001.
- [6] J. E. Wieselthier, G. D. Nguyen, and A. Ephremides, *On the Construction of Energy-efficient Broadcast and Multicast Trees in Wireless Networks*, IEEE Infocom, p585-594, 2000.
- [7] J. Wu, F. Dai, M. Gao, and I. Stojmenovic, *On Calculating Power-aware Connected Dominating Sets for Efficient Routing in Ad Hoc Wireless Networks*, Journal of Communications and Networks, Vol 4, Number 1, p59-70, March 2002.