

Construction of Power Efficient Routing Tree for Ad Hoc Wireless Networks using Directional Antenna

Qing Dai and Jie Wu

Department of Computer Science and Engineering
Florida Atlantic University
Boca Raton, Florida 33432
{qdai, jie}@cse.fau.edu

Abstract—In ad hoc wireless networks, nodes are typically powered by batteries. Therefore saving energy has become a very important objective, and different algorithms were proposed to achieve power efficiency during the routing process. Directional antenna was suggested to further decrease transmission energy as well as to reduce interference. In this paper, we discuss three algorithms for routing tree construction that take advantage of directional antenna, i.e, Simple-Linear (SL), Linear-Insertion (LI), and Reverse-Cone-Pairwise (RCP). Their performances are compared through a simulation study.¹

Keywords: *ad hoc wireless network, directional antenna, energy-efficient routing, transmission power*

I. INTRODUCTION: BACKGROUND AND RELATED WORK

Energy efficiency is an important consideration for ad hoc wireless networks, where nodes are typically powered by batteries. It is directly correlated with network longevity and connectivity, therefore affecting network throughput. Among all different components of power consumption, transmission cost appears to dominate, compared to receiving cost and computation cost. It has been shown that the power threshold p for a source node to reach its destination node is positively correlated to the distance between them, and can usually be expressed as $p = r^\alpha$, where d is the distance between the two nodes, and α is between 2 and 4. In previous studies, different metrics have been used, some measure the overall energy consumption of the network, while some others try to extend lifespan of individual nodes. The Broadcast Incremental Power algorithm (BIP) is a centralized algorithm attempting to minimize the overall energy in route determination [1]. It is similar to Prim's Minimum Spanning Tree algorithm, in that at any time all reached nodes form a single-rooted tree. Each step adds the node with the minimum incremental cost, calculated as the additional energy for it to be reached by any node within the tree, either by increasing power of a transmitting node, or by making a non-transmitting node transmit at a specific power level. It has already been proved that BIP has a constant approximation ratio of between 6 and 12, compared to the optimal solution [4]. On the other hand,

¹The work was supported in part by a grant from Motorola Inc. and NSF grants CCR 0329741, ANI 10073736, and EIA 0139806.

Algorithm I: Reverse-Cone-Pairwise (RCP)

```
1  unreachedNodes ← {all nodes except sourceNode}
2  reachedNodes ← {sourceNode}
3  edgesInGraph ← ∅
4  totalCost ← 0
5  while unreachedNodes ≠ ∅
6    for  $j \leftarrow 0$  to  $\text{size}[\text{reachedNodes}] - 1$ 
7      do find minNode  $i$  in unreachedNodes with
          minimum incrementalCost( $i, j$ )
8      reachedNodes ← reachedNodes ∪ minNode
9      unreachedNodes ← unreachedNodes − minNode
10     edgesInGraph ← edgesInGraph ∪ {edge( $i, j$ )}
11     totalCost ← totalCost + incrementalCost( $i, j$ )
```

Fig. 1. Algorithm I: Reverse-Cone-Pairwise

some power-aware routing approaches select routes that avoid nodes with low remaining power, which can be either absolute or relative power level. Different metrics may well lead to different algorithms, but they can also be combined to achieve a balance, thus optimizing overall energy consumption without depleting crucial nodes [5][2].

Recently, the use of directional antenna was proposed in order to reduce interference and to further increase power efficiency, because ideally, power consumed in case of directional antenna is only

$$r^\alpha \frac{\theta}{2\pi},$$

where θ is the beam angle. In one of the session-based studies, two algorithms were proposed: RB-BIP (Reduce Beam BIP) and D-BIP (Directional BIP). The former simply adds a beam-reducing step to the original BIP algorithm, so that each node can now transmit at its smallest possible angle. The latter incorporates the use of directional antenna at each step of the tree construction, i.e, a node in the tree could also increase its current transmission beam angle or shift the existing beam to reach a new destination node, in addition to increasing its transmission power, whichever gives the lowest incremental cost [2].

In our study, we try to find different centralized routing

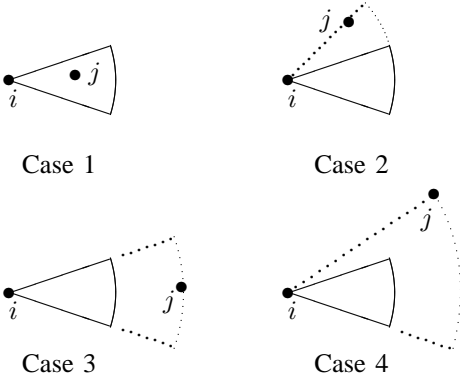


Fig. 2. Incremental transmission cost.

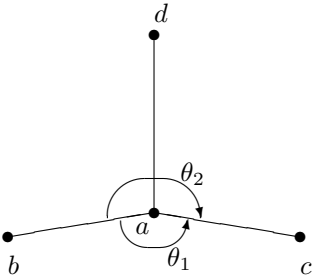


Fig. 3. Illustration of Reverse-Cone method.

algorithms for broadcasting with the use of directional antenna. The paper is organized as follows: three algorithms are introduced in Section II. We then simulate their performance with randomly generated networks in Section III, and discuss their applications and potentials in Section IV.

II. ALGORITHMS

Our objective is to find routing algorithms that are power efficient using directional antenna. In this study, we assume one beam for each node, and $\alpha = 2$ to calculate the transmission energy. Here we propose three algorithms, assuming global knowledge of node locations, adjustable transmission range and adjustable beam angle.

The first algorithm is a refinement of D-BIP [2], where the node with the minimum incremental cost is added at each step, while taking the adjustable antenna beam width into consideration (see Figure 1). The actual beam angle that is used has to exceed a minimum beam angle $minAngle$. All nodes that have already been reached can act as transmitting node, and all the non-tree nodes are potential destination nodes. Each possible transmitting-destination node pair is examined to determine the minimum power and beam angle needed to add a new destination node. When a transmitting node is adding a new destination node, if the new node does not fall into its current transmission beam, it can either shift or expand its previous beam to cover the new node. The incremental

```

MinBeam (srcNode, destNode)
1  calculate destAngle
2  insert destAngle into angleList
3  maxCone  $\leftarrow$  0
4  minBeam  $\leftarrow$  (beamStart  $\leftarrow$  0, beamEnd  $\leftarrow$  0)
5  for i  $\leftarrow$  0 to size[angleList]
6    coneAngle  $\leftarrow$  angleList[i + 1] - angleList[i]
7    if coneAngle > maxCone
8      maxCone  $\leftarrow$  coneAngle
9      minBeam  $\leftarrow$  (angleList[i + 1], angleList[i])
10 return minBeam

```

Fig. 4. Reverse-Cone method to find the minimum beam.

power is determined by subtracting the energy of the previous transmission beam from the new power, calculated as

$$incrementalCost(i, j) = (p' \times \theta' - p \times \theta) / (2 \times \pi),$$

where p' and θ' represent the new transmission power and beam angle, respectively, while p and θ are the previous power and beam angle. This process is demonstrated in Figure 2, where i and j are the source and destination nodes, respectively, and the cone represents the current transmission beam. In case 1, the destination node d could be enclosed within the current beam, including through a beam shift. It will be covered without any incremental cost. Otherwise, either the transmission beam has to be expanded (case 2) or transmission power needs to be increased (case 3), or both (case 4) for d to be reached. It could therefore be very costly.

During this process, it is desirable to use the minimum transmission beam angle whenever possible. A simple heuristic method to calculate the new beam is to keep the start and end point of the previous beam, and to expand either end that leads to a smaller increase of the overall beam span. We will see that this heuristic does not always provide the optimal beam angle. An example is illustrated in Figure 3. At first, the source node a is transmitting to two downstream neighbors b and c with a beam angle θ_1 . When a new destination node d joins, and when θ_1 is close to π , neither expanding beam to $b-c-d$ nor to $c-b-d$ provides the minimum beam. On the other hand, $b-d-c$ is the optimal solution in this case with a beam angle θ_2 . The Reverse-Cone method (Figure 4) is developed to calculate the minimum beam span. Instead of merely keeping the start and end point of the previous beam, each node keeps angle positions, calculated as the radius of a destination node from itself, of all its destination nodes in a sorted list *angleList*. The 2π circle around a transmitting node is then divided into several cones, each defined by the two adjacent neighbors in the *angleList*. Whenever adding a new destination node, the transmitting node will first insert the angle location of the new node into its *angleList*. The *angleList* is traversed to find the largest cone. The minimum new beam to cover all the nodes in *angleList* would be the reverse of this largest cone.

For the previous example in Figure 3, node a first adds nodes b and c in its sorted *angleList*. Of the two cones, $c-b$ counterclockwise is larger than $b-c$, therefore the reverse of cone $c-b$, i.e., cone $b-c$, or θ_1 , is selected as the minimum

Algorithm II: Simple Linear (SL)

```

1  $unreachedNodes \leftarrow \{\text{all nodes except } sourceNode\}$ 
2  $reachedNodes \leftarrow \{sourceNode\}$ 
3  $listHead \leftarrow sourceNode$ 
4  $totalCost \leftarrow 0$ 
5 while  $unreachedNodes \neq \emptyset$ 
6   for  $i \leftarrow 1$  to  $size[unreachedNodes] - 1$ 
7     do find  $minNode$  with minimum cost  $minCost$ 
8      $reachedNodes \leftarrow reachedNodes \cup minNode$ 
9      $listHead \leftarrow minNode$ 
10     $unreachedNodes \leftarrow unreachedNodes - minNode$ 
11     $totalCost \leftarrow totalCost + minCost$ 

```

Fig. 5. Algorithm II: Simple Linear

beam. When node a adds a new destination d , d is then inserted into a 's angle list, which now becomes $d-b-c$. After traversal of the new *angleList*, $b-c$ is found to be the largest cone, therefore the reverse of it, $c-d-b$, or θ_2 will be used as the new transmission beam.

The RCP algorithm generally provides good performance, except in the situations when the previous transmission beam was small, adding a new node at a distant angle position can be very expensive. Two other heuristic algorithms are therefore developed: Simple-Linear (SL) and Linear-Insertion (LI). In both of them, each transmitting node always uses the minimum angle $minAngle$ to reach exactly one downstream node, and there is only one downstream destination node for each transmitting node. As a result, a linear chain will be formed step by step, starting from the source node. Initially, the *reachedNodes* set includes *sourceNode* only, and the *unreachedNodes* set includes all other nodes. In SL (see Figure 5), initially the source node is the *listHead*, then at each step, a *minNode* is determined as an *unreached* node closest to the current *listHead*, and is added to the *reachedNodes* set to become the new *listHead*. This *listHead* is the only possible transmitting node to reach the next new node, until the *unreachedNodes* set is empty. In LI (Figure 6), an additional backtrack step is included when each node is added to the *reachedNodes* set. During the backtrack, after the *minNode* is determined, it is first tested for possible insertions into each position between any two adjacent nodes within the existing linear chain. The *insertionCost*, the incremental cost for inserting the new node, is calculated to check whether any insertion causes a saving of energy, compared to directly adding the *minNode* as the new *listHead*. If so, an insertion will take place where the *insertionCost* is the minimum. In this case, the previous *listHead* remains unchanged. If not, *minNode* would be attached as the new *listHead*. In the example in Figure 7, s represents the *sourceNode*, d_1 through d_6 are the destination nodes to be reached. The Linear-Insertion algorithm would add d_6 between s and d_1 in the backtrack process, which leads to a lower overall energy cost. While LI seems to be a better solution than SL, unfortunately, it does not always outperforms SL due to its heuristic nature (examples not shown).

Algorithm III: Linear-Insertion (LI)

```

1  $unreachedNodes \leftarrow \{\text{all nodes except } sourceNode\}$ 
2  $reachedNodes \leftarrow \{sourceNode\}$ 
3  $listHead \leftarrow sourceNode$ 
4  $totalCost \leftarrow 0$ 
5 while  $unreachedNodes \neq \emptyset$ 
6   for  $i \leftarrow 0$  to  $size[unreachedNodes] - 1$ 
7     do find  $minNode$  with minimum cost  $minCost$ 
8     for  $j \leftarrow 0$  to  $size[reachedNodes] - 1$ 
9       do insert  $minNode$  after  $reachedNode[j]$ 
10      find minimum  $InsertCost$ 
11      if  $InsertCost < minCost$ 
12        do  $minCost \leftarrow minInsertCost$ 
13        insert  $minNode$ 
14      else
15        do  $listHead \leftarrow minNode$ 
16       $reachedNodes \leftarrow reachedNodes \cup minNode$ 
17       $unreachedNodes \leftarrow unreachedNodes - minNode$ 
18       $totalCost \leftarrow totalCost + minCost$ 

```

Fig. 6. Algorithm III: Linear Insertion

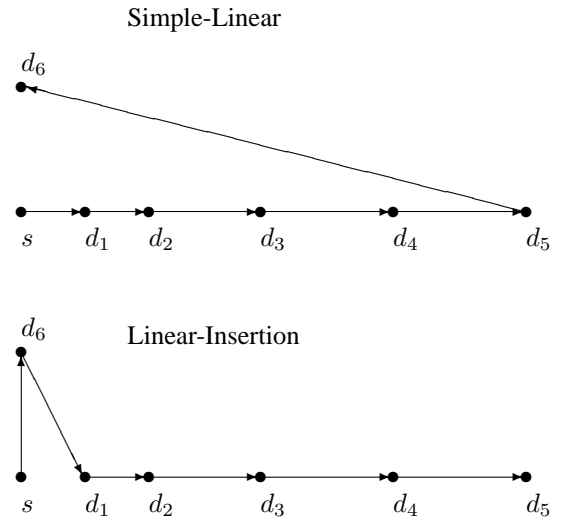


Fig. 7. Illustration of Linear-Insertion.

The complexity of the three algorithms are as follows: SI and LI have the similar complexity level of $\theta(n^2)$. RCP has a higher complexity of $\theta(n^4)$, with the Reverse-Cone method. To evaluate the relative performance of the three algorithms, simulations are performed on random networks. The result is shown in Section III.

III. SIMULATION RESULT AND CONCLUSION

We generate random network instances, and the three algorithms are used to obtain routing graphs, respectively. Their transmission costs are calculated and compared with each other.

Our results show that in most occasions, RCP produces routing solutions with good energy-efficiency. Between SL and

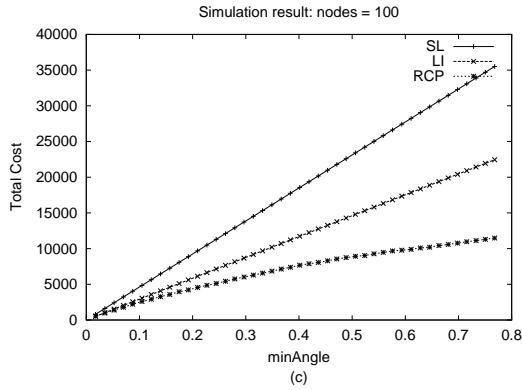
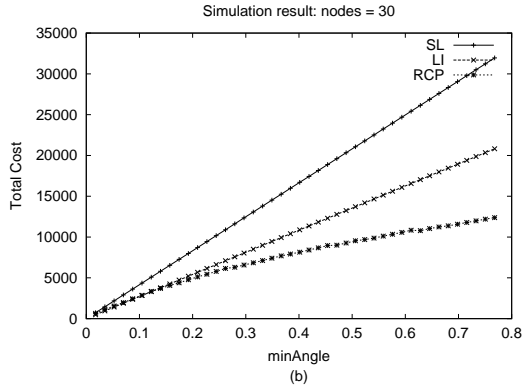
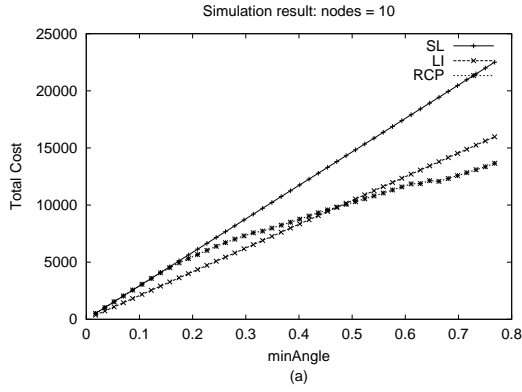


Fig. 8. Simulation results with $\min\text{Angle}$ from $\pi/90$ to $\pi/4$. Number of nodes is 10 (a), 30 (b) and 100 (c), respectively, in a fixed area.

LI, LI has a better performance. In simulation with different network density, we observed that when node distribution is sparse, LI outperforms RCP when the $\min\text{Angle}$ is relatively small. In the example shown in Figure 8, we set the number of nodes from 10 to 100 in the same size of area. As the $\min\text{Angle}$ increases gradually, we can see that energy costs of SL and LI routing increase with the $\min\text{Angle}$ linearly. This is because in SI and LI algorithms, the route construction process is not affected by the $\min\text{Angle}$, and the same routing graph will always be produced just as omnidirectional antenna is provided. The final cost is simply calculated by multiplying the original overall cost by a factor of $\min\text{Angle}/(2\pi)$. The result of RCP is close to SI when the $\min\text{Angle}$ is small,

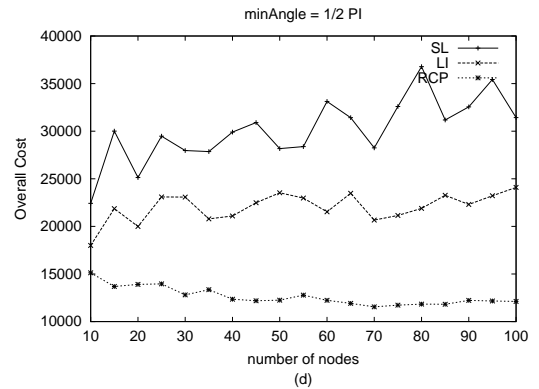
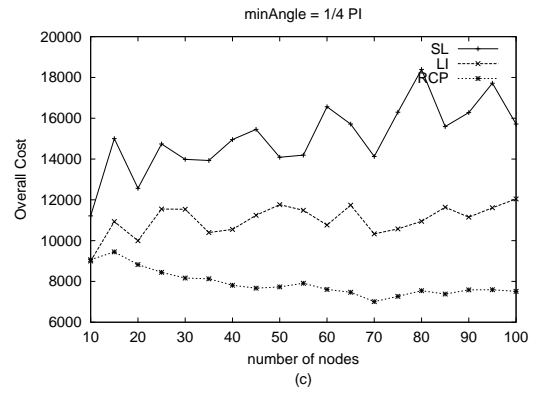
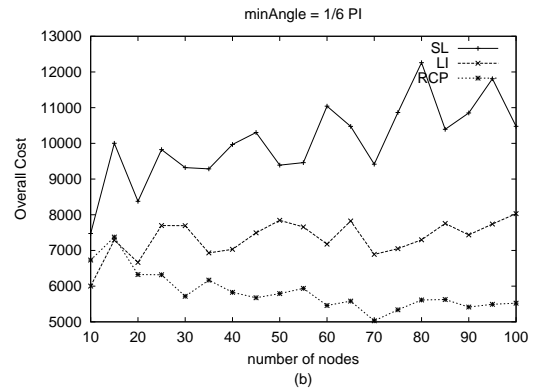
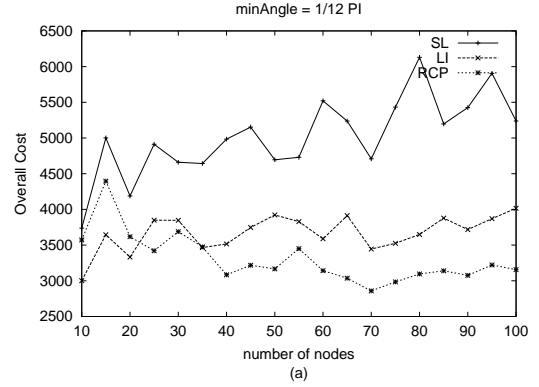


Fig. 9. Simulation results with number of nodes from 10 to 100 in a fixed area. $\min\text{Angle}$ is $\pi/12$ (a), $\pi/6$ (b), $\pi/4$ (c), and $\pi/2$ (d), respectively.

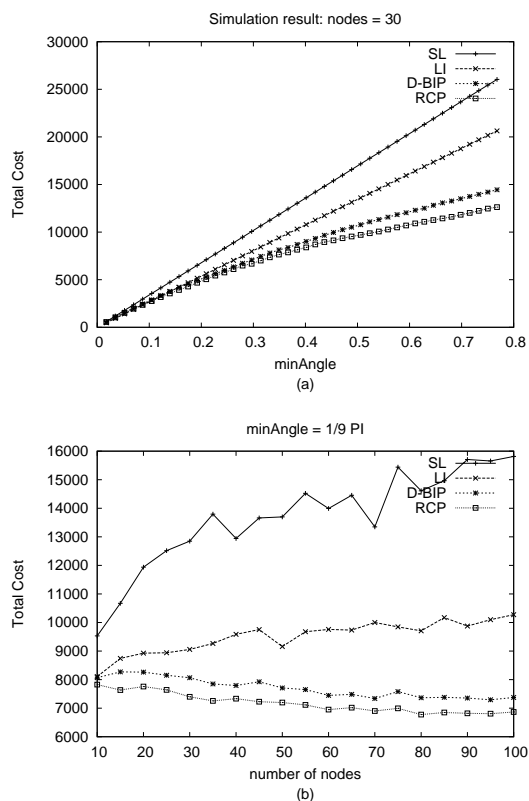


Fig. 10. Simulation results of SL, LI, and RCP compared with D-BIP. $minAngle$ from $\pi/180$ to $\pi/4$, with 30 nodes in a fixed area (a), and number of nodes from 10 to 100 in a fixed area, with $minAngle$ $\pi/9$ (b).

which makes sense since in this case RCP would most likely generate the same routing graph as SL. When $minAngle$ increases to a certain point, RCP will outperform SL, and eventually outperform LI, when shifting or expanding current beam shows advantage.

Simulations are also performed to measure the overall cost over the number of nodes. In a fixed area, the costs of SL and LI increment along with the increase of nodes, but the cost of RCP decreases, except when the node number and $minAngle$ are very small (Figure 9). It seems that in most cases, RCP has the best approximation ratio among the three.

To compare the performance of our three algorithms, especially RCP, with that of D-BIP, a simplified version of D-BIP is implemented. Same as before, we consider only the transmission cost, which is a function of the distance. As expected, RCP shows better energy efficiency than D-BIP (Figure 10).

Nevertheless, all three algorithms described above are greedy algorithms and heuristic in nature, with none being optimal. In fact, we can find special network instances between SL, LI or RCP, where one outperforms the other two (examples not shown). From the simulation result, it seems that when nodes are relatively sparse and $minAngle$ is small, LI is the best choice. Not only does it give the lowest routing cost, but its complexity is lower than RCP also. When the $minAngle$ increases to a certain extent or when network is dense, ex-

panding current beam to include additional destination nodes would be more cost efficient, and at this point, RCP provides the best performance. Currently, we are further analyzing their performance statistically. In the future, it will be interesting to know the approximation ratio of the three algorithms.

REFERENCES

- [1] J. E. Wieselthier, G. D. Nguyen, A. Ephremides, *On the construction of energy-efficient broadcast and multicast trees in wireless networks*, Proceedings of IEEE INFOCOM 2000, p585-594.
- [2] J. E. Wieselthier, G. D. Nguyen, A. Ephremides, *Energy-limited wireless networking with directional antennas: the case of session-based multicasting*, IEEE INFOCOM 2002, p190 -199.
- [3] L. Li, J. Y. Halpern, P. Bahl, Y. -M. Wang, R. Wattenhofer, *Analysis of a cone-based distributed topology-control algorithm for wireless multi-hop networks*, Proceedings of ACM Principles of Distributed Computing, p264-273, 2001.
- [4] P. -J. Wan, G. Calinescu, S. -Y. Li, O. Frieder, *Minimum-energy broadcasting in static Ad Hoc wireless networks*, Wireless Networks, Vol. 8, p607-617, 2002.
- [5] J. Wu, F. Dai, M. Gao, I. Stojmenovic, *On calculating power-aware connected dominating set for efficient routing in ad hoc wireless networks*, IEEE/KICS Journal of Communications and Networks, Vol. 4, p59-70, March 2002.