Analysis of Minimum-energy Path-preserving Graphs for Ad-hoc Wireless Networks*

Ashikur Rahman
Mahmuda Ahmed
Shobnom Zerin
Department of Computer Science and Engineering,
Bangladesh University of Engineering and Technology,
Dhaka-1000, Bangladesh
ashikur@cs.ualberta.ca, {mahricky, shobnom_zerin}@yahoo.com

We consider *ad-hoc* wireless networks and the topology control problem defined as minimizing the amount of power needed to maintain connectivity. The issue boils down to selecting the optimum transmission power level at each node based on the position information of reachable nodes. Local decisions regarding the transmission power level induce a subgraph of the maximum powered graph $G_{\max}$ in which edges represent direct reachability at maximum power. In this paper we propose an analysis for constructing minimum-energy path-preserving subgraphs of $G_{\max}$, i.e. subgraphs minimizing the energy consumption between node pairs. We also propose an algorithm for constructing subgraph of $G_{\max}$ based on one-hop neighbor information. By presenting experimental results we show the effectiveness of our proposed algorithm.

Keywords: *ad-hoc* network, topology control, energy efficiency

1. Introduction

Minimizing energy consumption is a major design goal in *ad-hoc* wireless networks. If networks are designed by preserving minimum energy paths, then a significant reduction in energy consumption can be achieved. We have to choose the minimum power level for each node to minimize the energy consumption for the whole network. We use the same model as [1], which considers an n-node, multi-hop, *ad-hoc*, wireless network deployed on a two-dimensional plane. Suppose that each node is capable of adjusting its transmission power up to a maximum denoted by $P_{\max}$. Such a network can be modeled as a graph $G_{\max} = (V, E)$, with the vertex set V representing the nodes, and the edge set defined as follows:

$$E = \{\langle x, y\rangle \mid \langle x, y\rangle \in V \times V \wedge d(x, y) \leq R_{\max}\} \quad (1)$$

where $d(x, y)$ is the distance between nodes x and y and $R_{\max}$ is the maximum distance reachable by a transmission at the maximum power $P_{\max}$. The graph $G_{\max}$ defined in this way is called the *maximum powered network*. By setting a reduced power level for each node, messages can be transmitted with a minimum energy among all of the paths in the network. This will lessen the average node degree and produce a subgraph of $G_{\max}$, while maintaining global connectivity.

We say that a graph $G' \subseteq G_{\max}$ is a minimum-energy path-preserving graph or, alternatively, that it has the *minimum energy property*, if for any pair of nodes (u, v) that are connected in $G_{\max}$, at least one of the (possibly multiple) minimum-energy paths between u and v in $G_{\max}$ also belongs to G'. Minimum-energy path-preserving graphs were first defined in [2]. Typically, many minimum-energy path-preserving graphs can be formed from the original graph $G_{\max}$. It has been shown that the smallest of such subgraphs of $G_{\max}$ is the graph $G_{\min} = (V, E_{\min})$, where $(u, v) \in E_{\min}$ if and only if there is no path of length greater than one from u to v that costs less energy than the energy required for a direct transmission between u and v. Let $G_i = (V, E_i)$ be a subgraph of

SIMULATION, Vol. 85, Issue 8, August 2009 511–523

DOI: 10.1177/0037549709340731

* A preliminary version of this paper has been published in the *International Symposium on Performance of Computer and Telecommunication Systems (SPECTS 2008).*

$G_{\max} = (V, E)$ such that $(u, v) \in E_i$ if and only if $(u, v) \in E$ and there is no path of length i that requires less energy than the direct one-hop transmission between u and v. Let $|V| = n$. Then $G_{\min}$ can be formally defined as follows:

$$G_{\min} = G_2 \cap G_3 \cap G_4 \cap \ldots \cap G_{n-1}. \quad (2)$$

It is easy to see that any subgraph G' of $G_{\max}$ has the *minimum-energy property* if and only if $G' \supseteq G_{\min}$. Thereby, each of $G_i \supseteq G_{\min}$, for any $i = 2, 3, \ldots, n-1$ is a *minimum-energy path-preserving graph*.

Here G_2 is the first of the series G_i, it is the most practically useful derivative of the maximum powered graph $G_{\max}$. An algorithm for constructing one such graph, G_2, was presented in [2]. It works reasonably well in dense networks but its performance degrades considerably when the network density drops to medium or low. Construction of G_2 efficiently in sparse and moderately dense networks with some assistance of the Medium Access Control (MAC) layer was presented in [1] and [3].

Not surprisingly, construction of high-order subgraphs (i.e. G_3, G_4, etc.) even demands more hop neighborhood information. For example, to construct G_3 we need two-hop information that means information from the immediate neighbors and the neighbors of those immediate neighbors. The number of links in G_2 and G_3 is considerably less compared with the number of links within the maximum powered network. Interestingly, the number of links in G_3 is not reduced significantly compared with G_2 but G_3 requires one hop more information than G_2. On the other hand, if we can construct G_3 and G_2 in a distributed way, we can easily construct $G_2 \cap G_3$ in the same fashion. The subgraph, $G_2 \cap G_3$ is definitely closer to $G_{\min}$ than G_2 or G_3. As the construction of G_3 requires one-hop and two-hop neighborhood information, consequently it involves exchanging many messages, and thus it causes obvious energy overheads. In this work, one of our major goals is to construct a subgraph of a *maximum powered network* which has less links than G_2 but at the same time can be constructed easily without increasing the overhead. To achieve this goal, at first we construct G_2 and a close approximation of G_3 based on one-hop neighbors found within the region having radius $R_{\max}$. Then we take intersection of these two subgraphs (i.e. G_2 and close approximation of G_3). Therefore, instead of constructing the exact subgraph $G_2 \cap G_3$ (which requires two-hop neighborhood information), we construct a close approximation to $G_2 \cap G_3$ collecting only one-hop neighborhood information. The resulting graph, called a *Local Minimum-energy Communication Network* (LMECN), is nothing but $G_2 \cap G_3$ with limited information. We present here a distributed algorithm to locally construct close approximation of $G_2 \cap G_3$. This approach requires no more information than G_2 but produces a reasonable amount of savings. A preliminary version of this paper has been published in [4]. However, in this paper, through extensive simulation results we show the effectiveness of our proposed distributed algorithm to locally construct $G_2 \cap G_3$.

2. Related Works

Recent research has lead to the exposure of mainly three dimensions directed at power control algorithms for wireless mobile networks. The first class of solutions looks at the issue from the perspective of the MAC layer. Monks et al. [5] propose a modification to the IEEE 802.11 Request to Send–Clear to Send (RTS-CTS) handshake whereby a node overhearing a CTS packet uses the received power to estimate its distance to the sender. Also, Sing and Raghavendra [6] propose techniques for powering-off the transceivers when they are not active.

The significance of the second approach lies in its dealing with the network layer. In this approach, the routing problem and the power control problem are jointly solved and termed as *power-aware routing*. Most of the schemes in this class use distributed variants of the Bellman–Ford algorithm with various flavors of cost metrics derived from the notion of power. Metrics taken as standard are as follows [7]: energy consumed per packet, time to network partition, variance in node power levels, and (power) cost per packet.

The findings in the second approach leads to the third approach that separates routing from topology control. Topology control protocol is assumed to be run under routing protocol as a separate independent layer and its purpose is to create a platform for power saving. Ramanathan and Rosales-Hain [8] describe two centralized heuristic algorithms to minimize the maximum transmission power while trying to maintain connectivity or bi-connectivity, called *Local Information No Topology* (LINT) and *Local Information Link-State Topology* (LILT).

COMPOW proposed by Narayanaswamy et al. [9], chooses the smallest common power level for each node emphasizing mainly: (1) connectivity, (2) traffic carrying capacity maximization, (3) contention reduction in the MAC layer, and (4) low-power requirement. It is preassumed that the distribution of nodes are homogeneous, which in the long run may appear as a serious flaw. CLUSTERPOW [10] was designed to overcome the shortcomings of COMPOW by introducing node clusters. However, it still suffers from a significant message overhead.

Cone-based Topology Control (CBTC) [11], generates a graph structure similar to that proposed by Chen et al. [12]. The basic variant of CBTC takes a parameter α, and each node u determines a power level $p_{u,\alpha}$ such that in every cone of degree α surrounding u, there is at least one node v reachable by u at $p_{u,\alpha}$. Each node starts with an initial small transmission power and gradually increases the power until the above condition is satisfied. Then the final graph G_α contains all edges uv constructed that way. The authors have proved that if $\alpha \leq 5\pi/6$, the resultant

graph is connected, provided that the original ($G_{\max}$) was. A serious drawback of the algorithm is the need to decide on the suitable initial power level and the increment at each step. The choice of these two parameters may have a significant impact on the number of overhead messages needed to create the final topology.

Rodoplu and Meng [13] introduce the notion of *relay region* based on a specific power model. Their algorithm was later modified by Li and Halpern [2] to trim some unnecessary edges. Our work closely relates to these two studies.

Li et al. [14] propose a distributed topology control algorithm (called LMST) based on constructing minimum spanning trees. Their algorithm achieves three goals: (1) connectivity, (2) bounded node degree (at most six) and (3) bi-directional links. In LMST, each node u collects the position information of all neighbors reachable at the maximum power. Based on this information, u creates its own local minimum spanning tree among the set of neighbors, where the weight of an edge is the necessary transmission power between its two ends. Once the tree has been constructed, u contributes to the final topology those nodes that are its neighbors in the spanning tree. One problem with LMST is that the resulting graph does not preserve the minimum-energy paths.

3. Algorithm

3.1 Power Model

We are using the same power model as [1]. Here we assume the well-known, generic, two-way, channel path loss model, where the minimum transmission power is a function of distance [15]. To send a packet from node x to node y, separated by a distance $d(x,y)$, the minimum necessary transmission power is approximated by

$$P_{\text{trans}}(x, y) = t \times d^{\alpha}(x, y) \tag{3}$$

where $\alpha \geq 2$ is the *path loss factor* and t is a constant. Signal reception is assumed to cost a fixed amount of power denoted by r. Thus, the total power required for one-hop transmission between x and y becomes

$$P_{\text{total}}(x, y) = t \times d^{\alpha}(x, y) + r \tag{4}$$

The model assumes that each node is aware of its own position with a reasonable accuracy, e.g. via a Global Positioning System (GPS) device.

3.2 Previous Approach to Constructing G_2

An algorithm was presented in [13] based on *relay region*. The concept is as follows.

Given a node u and another node v within u's communication range (at $P_{\max}$), the relay region of node v as perceived by u (with respect to u), $R_{u\to v}$, is the collection of points such that relaying through v to any point in $R_{u\to v}$ takes less energy than a direct transmission to that point (see Figure 1)).

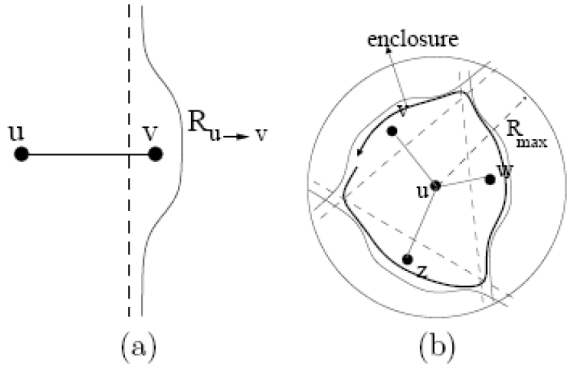

Figure 1. (a) The relay region of v with respect to u; (b) the enclosure of node u

Given the definition of a relay region, the algorithm for constructing G_2 becomes straightforward. The construction of G_2 efficiently in sparse and moderately dense networks with some assistance of the MAC layer was presented in [1].

3.3 Algorithm

We propose an algorithm for constructing close approximation of $G_2 \cap G_3$ with much less overhead which works in a distributed way (Algorithm 1). We refer to our algorithm as the *Local Minimum-energy Communication Network* (LMECN) *algorithm* or throughout this paper. In the LMECN algorithm, function $cost(x, y)$ takes two nodes x and y as parameters and calculates the cost of direct transmission between them according to Equation (4) assuming that t, the predetection threshold, is equal to 10^{-7} mW, the additional receive power $r = 20$ mW and that the *path loss factor* $\alpha = 4$. The function getCommonNeighbor(x, y, $\aleph$) determines the common neighbor set of x and y having information of set of nodes $\aleph$.

The operation of the above algorithm is described from the viewpoint of one node s. Here s broadcasts a single Neighbor Discovery Message (NDM) at the maximum power $P_{\max}$. While s collects the replies of its neighbors, it learns their identities and locations. It constructs the sets $\aleph$ and $\aleph_s$. Initially, all of those sets are empty (s does not even know what neighbors there are). Here $\aleph$ is defined as the set of all nodes that s has discovered so far within its maximum transmission range and $\aleph_s$ is the set of its neighbors in LMECN. Node s has to maintain direct link or direct communication with each node in $\aleph_s$. It is easy to see that $|\aleph_s| \leq |\aleph|$.

When node s receives a reply from node r it is added to the neighbor set $\aleph_s$, then it considers two cases for

Algorithm 1 LMECN ALGORITHM

```
1: /* s is the node running this algorithm */
2: /* ℵ is the set of discovered node so far by s */
3: /* ℵ_s is the Neighbor set of node s generated by LMECN */
4: /* Suppose, s got a reply from node r */
5: ℵ_s = ℵ_s ∪ r
6: for each n in ℵ do
7:    /* get list of common neighbors between n and r */
8:    D =getCommonNeighbor(n, r, ℵ)
9:    /* get list of common neighbors between s and r */
10:   G =getCommonNeighbor(s, r, ℵ)
11:   Segment 1:
12:   if r is in ℵ_s then
13:      /* case 1(a): check for two length path between s and r relaying through n */
14:      if (cost(s, r) ≥ cost(s, n) + cost(n, r)) then
15:         ℵ_s = ℵ_s − {r}
16:         goto Segment 2
17:      end if
18:      /* case 1(b): check for three length path between s and r relaying through n and d */
19:      for each d in D do
20:         if (cost(s, r) ≥ cost(s, n) + cost(n, d) + cost(d, r)) then
21:            ℵ_s = ℵ_s − {r}
22:            goto Segment 2
23:         end if
24:      end for
25:   end if
26:   Segment 2:
27:   if n is in ℵ_s then
28:      /* case 2(a): check for two length path between s and n relaying through r */
29:      if (cost(s, n) ≥ cost(s, r) + cost(r, n)) then
30:         ℵ_s = ℵ_s − {n}
31:         goto Segment 3
32:      end if
33:      /* case 2(b): check for three length path between s and n relaying through r and d */
34:      for each d in D do
35:         if (cost(s, n) ≥ cost(s, r) + cost(r, d) + cost(d, n)) then
36:            ℵ_s = ℵ_s − {n}
37:            goto Segment 3
38:         end if
39:      end for
40:      /* case 2(c): check for three length path between s and n relaying through g and r */
41:      for each g in G do
42:         if (cost(s, n) ≥ cost(s, g) + cost(g, r) + cost(r, n)) then
43:            ℵ_s = ℵ_s − {n}
44:            goto Segment 3
45:         end if
46:      end for
47:   end if
48:   Segment 3: continue
49: end for
50: ℵ = ℵ ∪ r
```

each node in ℵ. First it tries to find a node n in ℵ such that $cost(s, r) >= cost(s, n) + cost(n, r)$, then r is discarded from $\aleph_s$ because node n can be used as a relay to transmit to r (see Figure 2(a)). If $cost(s, r) < cost(s, n) + cost(n, r)$, then there is no path of length two that can be used to transmit to r with less energy than

direct transmission between s and r. So, r will not be discarded from neighbor set $\aleph_s$. Thus, G_2 is forming here. Now a length-three path that costs less energy than a direct transmission between s and r is searched. If d is a common neighbor node between n and r then the condition $cost(s,r) >= cost(s,n) + cost(n,d) + cost(d,r)$ is checked. If there is a node that fulfills the above condition then r is discarded from $\aleph_s$ because there is path of length three which costs less than direct transmission (see Figure 2(b)).

Now our LMECN algorithm searches for the nodes that should be removed from neighbor set $\aleph_s$ after including r. First it discards all of the nodes for which r can be used as a relay in the case of a length-two path (see Figure 2(c)). Thus, G_2 is formed here. Then search for a node d in the common neighbor set of n and r and for a node g in the common neighbor set of s and r is done to find a length-three path that is less costly than direct transmission between s and n (see Figure 2(d) and (e)).

This algorithm constructs G_2 in a proper way and then tries to construct G_3 from available (partial) node information.

As we know that the construction of G_3 requires two-hop information from neighbors which causes more messages to be exchanged and increases energy overheads. So we tried to improve the performance of G_2 here. We constructed G_3 using the local information found by exchanging one-hop information as in the G_2 case. We refer to the graph found by LMECN as G_{LMECN} throughout this paper.

3.4 Why G_{LMECN} is Close to $G_2 \cap G_3$

In this section we present scenario where local information from immediate neighbors are not enough to construct G_3. In Figure 3 the circle represents $R_{\max}$ neighborhood of node a. Node b and node c are inside its neighbor region but node d is outside of its transmission range. Suppose node d is situated very close to $R_{\max}$ i.e., near the boundary of node a. As node a has only local information in LMECN, it will see that there is no two or three length path that costs less than direct communication between a and b, therefore it maintains link ab. But in actual case it might happen that relaying through node c and d may cost less than direct communication between a and b. So, $acdb$ is the three length path that is considered in G_3 but not in LMECN. This is a case where LMECN produces more link than $G_2 \cap G_3$.

Lemma 1. *We have* $G_{\text{LMECN}} \supseteq G_2 \cap G_3$.

Proof. A link between two nodes $u, v \in V$ is deleted in LMECN only when there is another minimum-energy path of length two or three between the nodes, i.e. LMECN applies G_2 and locally G_3, thus, it does not destroy any minimum-energy path having length two or three. In addition, LMECN produces some extra links as described in previous section. So the resultant graph G_{LMECN} is definitely a superset of $G_2 \cap G_3$. □

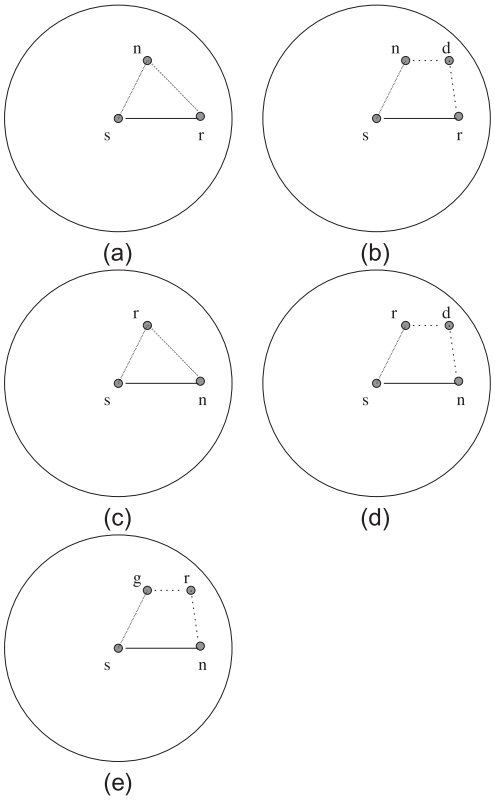

Figure 2. Illustrating LMECN algorithm with different scenarios. (a) Case 1(a) of the algorithm where a length-two path between s and r is searched. (b) Case 1(b) of the algorithm where a length-three path between s and r is searched. (c) Case 2(a) of the algorithm where a node n is searched for which node r can be used as a relay from s. (d) Case 2(b) of the algorithm where a node n is searched for which node r is located on a length-three path from s. (e) Case 2(c) of the algorithm where a node n is searched for which node r is located at the second hop on a length-three path from s.

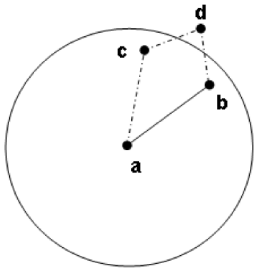

Figure 3. An extraneous edge

Table 1. Simulation parameters at a glance

Topic	Description
Deployment area	670 m × 670 m, 1,000 m × 1,000 m
Transmission range	250 m
Total number of nodes	25–300
Types of network	Square-shaped and C-shaped
Node distribution	Uniform, zipf and irregular density

Lemma 2. *LMECN preserves connectivity.*

Proof. We presented $G_{\min}$ as

$$G_{\min} = G_2 \cap G_3 \cap G_4 \cap \cdots \cap G_{n-1}$$

According to [2], $G_{\min}$ is the smallest subgraph with the minimum-energy property. He $G_{\min}$ has no redundant edges and also preserves connectivity. Thereby, each of $G_i \supseteq G_{\min}$, for any $i = 2, 3, \ldots, n-1$ is a minimum-energy path-preserving graph and also preserves connectivity. Here $G_2 \cap G_3$ is a superset of $G_{\min}$. According to Lemma 2 $G_{\text{LMECN}} \supseteq G_2 \cap G_3$ and, thus, preserves connectivity. □

4. Simulation Results

4.1 Topological Analysis

By simulation we evaluated the performance of our algorithm. Simulations were performed on several topologies. Topologies generated for the simulation can be broadly classified into two major divisions: (1) square-shaped networks; and (2) C-shaped networks. While generating square-shaped networks, we deployed 25–300 nodes over a flat square area of 670 m × 670 m and/or 1,000 m × 1,000 m, according to (1) a *uniform distribution* and (2) a *zipf distribution*. A description of each type of distribution is provided in the subsequent sections. For each category of topologies, we varied node density by slowly increasing total number of nodes present in the network on the same area. The transmission range of each node was set to 250 m for all simulation runs. Table 1 summarizes simulation parameters.

As a performance measure, we use the following two metrics.

(1) *Number of links*. We use total number of links present in the network as an indication of quality of any *minimum-energy path-preserving* algorithm. Obviously, an algorithm producing less number of links without loosing connectivity saves more energy.

(2) *Overhead*. Overhead is a measure of how costly an algorithm is. Here we count the total number of messages generated by an algorithm to produce its desired *minimum-energy path-preserving* graph. More overhead has a negative impact on energy savings.

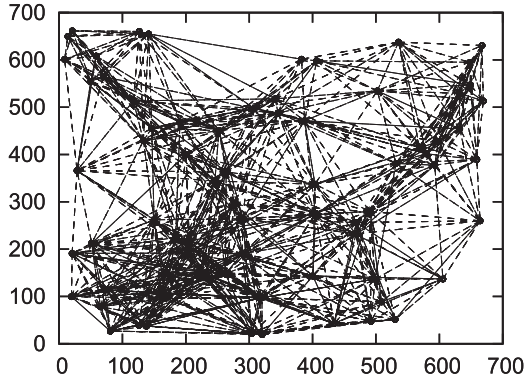

Figure 4. Original graph (uniform distribution) with 75 nodes

In all algorithms, a node s broadcasts NDM at first to find out about its neighbors. In the case of constructing G_2, the number of reply messages for any node s is equal to *Node Degree*(s). *Node degree* is defined as the number of neighbors it has within the region having radius $R_{\max}$. For constructing $G_2 \cap G_3$, the number of reply messages is equal to $NodeDegree(s) + \sum_{n \in neighbor(s)} NodeDegree(n)$, i.e. the node degree of its own and all of its neighbors.

4.2 Performance on Square-shaped Networks Under a Uniform Distribution

4.2.1 Sample Topological Graph

For these kinds of topologies we use random node placement, which yields a fairly uniform topology. A typical example of such topology is shown in Figure 4. It shows a typical uniform deployment of 75 nodes, each having a maximum communication range of 250 m. This is the starting graph $G_{\max}$ for our algorithm. The graph found by G_2 is Figure 5 which shows a large amount of savings over the original graph $G_{\max}$. The next graph shown in Figure 6 is found by G_3, and this also has a large amount of savings with respect to original graph $G_{\max}$ (as shown in Figure 4). So, both G_2 and G_3 shows good results. The graph that is closer to $G_{\min}$ than G_2 and G_3 is $G_2 \cap G_3$ as shown in Figure 7. This graph has the smallest number of links among the graphs mentioned above. Figure 8 shows the graph we found using our proposed LMECN algorithm. Here the graph has just one more link than that of $G_2 \cap G_3$. This implies that the performance of LMECN is much closer to $G_2 \cap G_3$ but incurs much less overhead

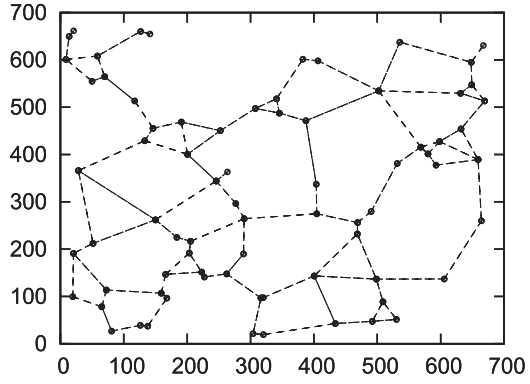

Figure 5. G_2 (uniform distribution) with 75 nodes

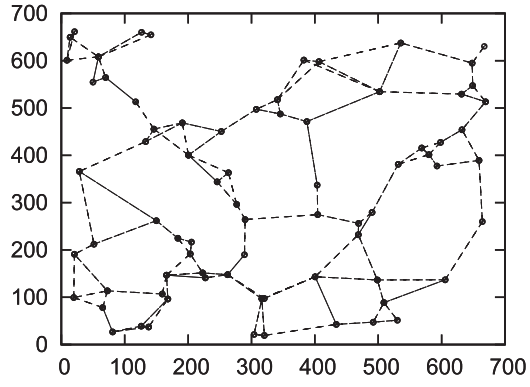

Figure 6. G_3 (uniform distribution) with 75 nodes

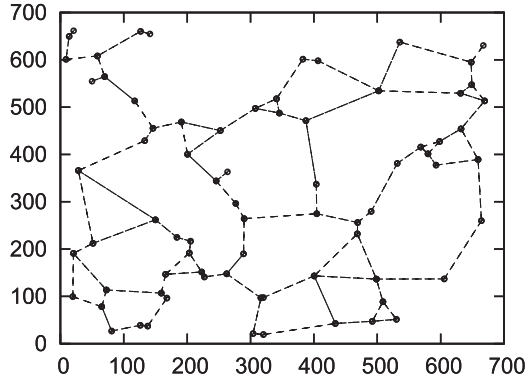

Figure 7. $G_2 \cap G_3$ (uniform distribution) with 75 nodes

than constructing G_3 (which requires two-hop neighbor information) to calculate $G_2 \cap G_3$.

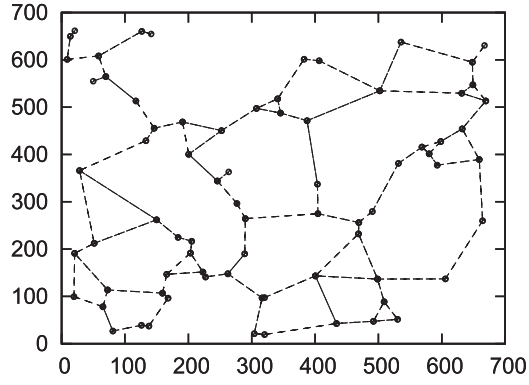

Figure 8. G_{LMECN} (uniform distribution) with 75 nodes

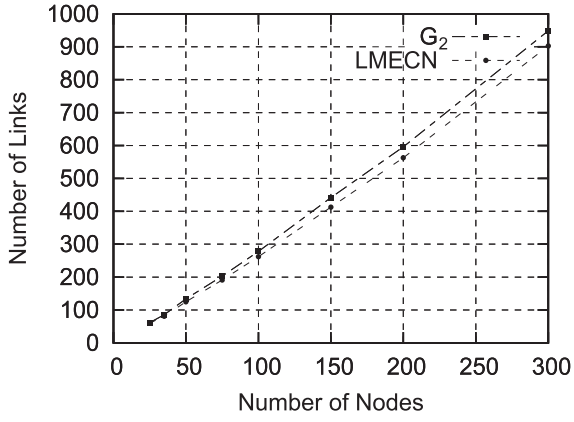

Figure 9. Comparison of the number of links between G_2 and G_{LMECN} under a uniform node distribution

4.2.2 Comparison of the Number of Links between G_2 and LMECN

Here we compare the number of links present in the graph produced by our proposed LMECN with the number of links present in G_2 constructed by using the algorithm proposed in [2]. For this purpose we first deploy 25–300 nodes over a flat square area of 670 m × 670 m according to a uniform distribution and simulate the network both for G_2 and LMECN. For illustration, Figure 9 shows a comparison between the number of edges in G_2 and in LMECN. For a specific number of nodes there are five different scenarios. We have taken average number of edges or links for each set. We found here a saving of up to 4.85% without increasing the overhead (see Table 2 for an overhead comparison).

Table 2. Overhead comparison between G_3 and G_{LMECN} under a uniform node distribution

Number of nodes	Overhead messages in G_{LMECN}	Overhead messages in G_3
25	160.8	1,342.4
35	334.8	3,910.8
50	810.4	15,613.6
75	1,637.2	41,130.8
100	3,082.0	106,442.8
150	6,978.8	361,218.0
200	12,421.6	852,938.4
300	27,777.6	2,821,319.2

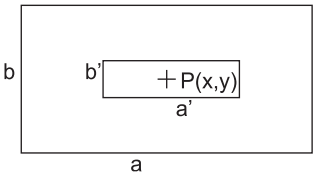

Figure 10. Illustrating zipf distribution

4.2.3 Overhead Analysis

Next, we took the number of reply messages as the overhead. To analyze the overhead cost incurred by the construction of G_3 and G_{LMECN} (or G_2) we simulate networks of 25–300 nodes deployed over a flat square area of 670 m × 670 m spread under a uniform distribution (Figure 4). For a specific number of nodes there are five different scenarios. We have taken the average number of overhead messages for each set. We found that construction of G_3 causes a huge amount of overhead with respect to G_{LMECN}. Table 2 shows a comparison of overhead between construction of G_3 and generated by the LMECN algorithm. More savings are achieved with denser networks.

4.3 Performance on Square-shaped Networks Under a Zipf Distribution

4.3.1 Sample Topological Graph

We also experimented with biased node distribution to see how our proposed algorithm behaves. Following the standard zipf bias, we assumed that 80% of nodes are distributed over 20% of the total deployment area and the remaining 20% nodes are distributed over the remaining 80% of the deployment area (the so-called 80–20 rule). For a more formal description, consider Figure 10. The total deployment area is the larger rectangular region with

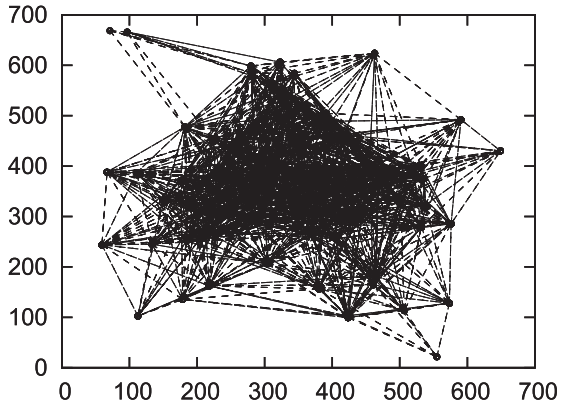

Figure 11. Original graph (zipf distribution) with 75 nodes

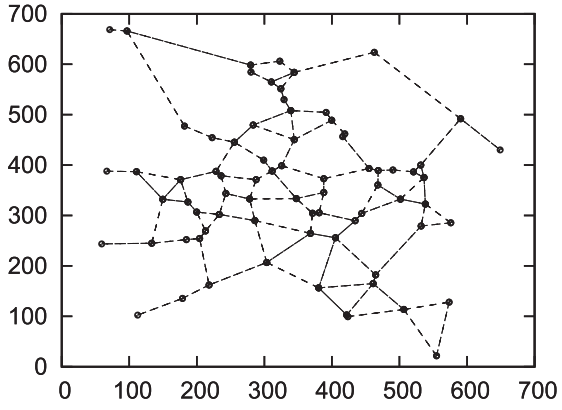

Figure 12. G_2 (zipf distribution) with 75 nodes

the dimensions a and b. Let P (the focal point of the distribution) be located at (x, y). The smaller rectangle centered at P has dimensions a' and b' such that

$$\frac{a'b'}{ab} = \alpha$$

where $\alpha = 0.2$. The network is generated in such a way that the probability of a node falling within the interior rectangle is $\beta = 1 - \alpha$, i.e. 0.8 in our case.

Figure 11 shows a typical topology reduction scenario involving 75 nodes under this biased distribution of nodes. The maximum communication range of each node was 250 m. The next two figures show the graph found by G_2 (Figure 12) and G_3 (Figure 13) where both shows good savings. Under a zipf distribution $G_2 \cap G_3$ (Figure 14) and G_{LMECN} (Figure 15) both show the same number of links. We have obtained the same result here without increasing the overhead which is definitely a good achievement.

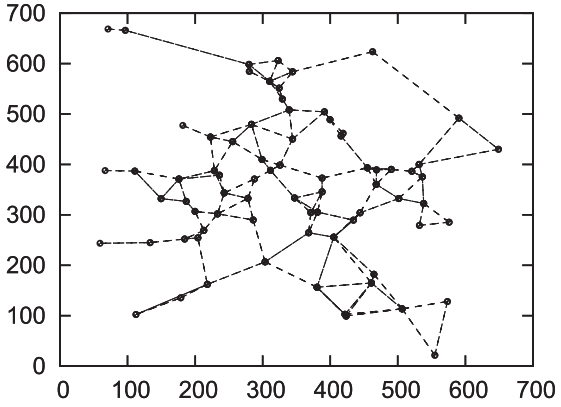

Figure 13. G_3 (zipf distribution) with 75 nodes

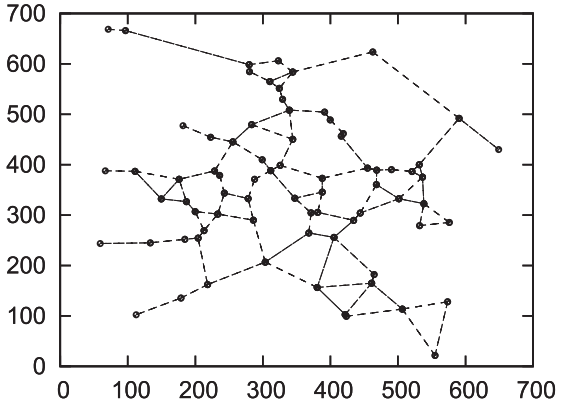

Figure 14. $G_2 \cap G_3$ (zipf distribution) with 75 nodes

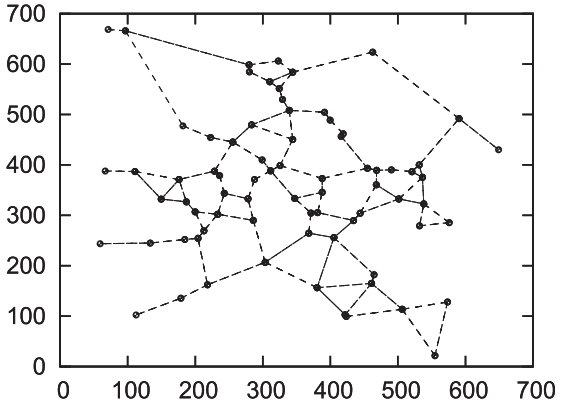

Figure 15. G_{LMECN} (zipf distribution) with 75 nodes

4.3.2 Comparison on Number of Links between G_2 and G_{LMECN}

Again, we simulate on a network of 25–100 nodes deployed over a flat square area of 670 m × 670 m according to a zipf distribution. The communication range was set to 250 m. Figure 16 shows a comparison between the number of edges in G_2 and in LMECN. Up to 6.18% of links is saved here. In summary, the savings are quite satisfactory and considerably higher for denser networks.

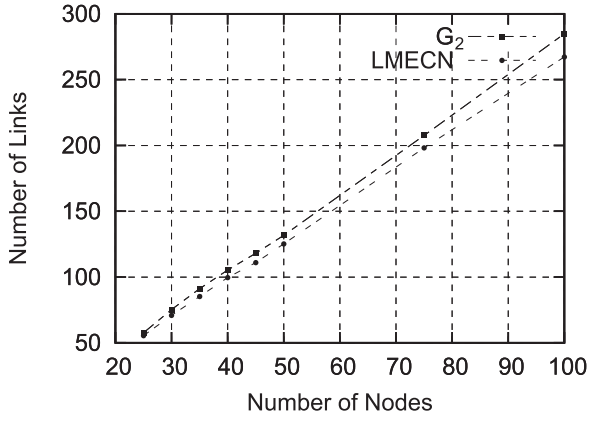

Figure 16. Comparison of the number of links between G_2 and G_{LMECN} under a zipf node distribution

Table 3. Overhead comparison between G_3 and G_{LMECN} in a zipf distribution

No of nodes	Overhead messages in G_{LMECN}	Overhead messages in G_3
25	308.8	4,799.6
30	357.6	5,274.4
35	515.2	9,506.4
40	675.2	13,656.4
45	930.8	23,031.2
50	1,146.4	24,828.4
75	2,592.4	104,965.6
100	4,377.2	221,260.8

4.3.3 Overhead Analysis

For the overhead analysis we performed similar sets of simulations as performed for uniform distributions. The networks of 25–100 nodes were deployed over a flat square area of 670 m × 670 m spread under a zipf-like distribution (Figure 10). Again, for a specific number of nodes there were five different scenarios. We have taken the average number of overhead messages for each set. We found that construction of G_3 causes a huge amount of overhead with respect to G_{LMECN}. Table 3 shows a summary of overhead comparison under different size networks.

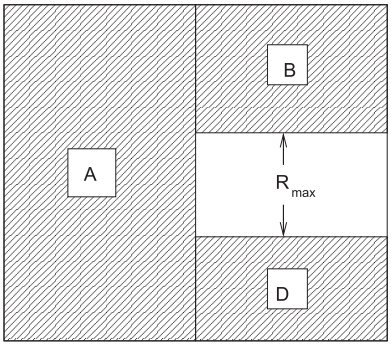

Figure 17. Illustration of the node deployment strategy to form C-shaped networks

4.4 C-shaped Network

4.4.1 Sample Topological Graph

Being motivated by the work done in [16] and [17], we also evaluated the performance of the proposed LMECN algorithm in C-shaped networks. The nodes are carefully placed in such a way so that the overall network topology looks like a letter 'C' in the English alphabet. The node deployment strategy to form a C-shaped network is described as follows: consider Figure 17 where a rectangular deployment area is divided into three sub-regions named A, B and D. The regions B and D are separated by a distance equal to the maximum possible transmission range of a node, i.e. $R_{\max}$ so that any node placed in region B are unreachable from any node placed in region D. Now the total number of nodes n is divided into three groups n_1, n_2 and n_3 in proportion to the ratio of the area A, B and D (i.e. $n_1 : n_2 : n_3 = A : B : D$ and $n_1+n_2+n_3 = n$). Then n_1, n_2 and n_3 nodes are uniformly placed within the rectangles A, B and D, respectively. The result is a C-shaped network. This kind of C-shaped network provides a good platform for testing the capabilities of algorithms in handling irregular topologies, where the hop distances between nodes can be very different from the actual Euclidean distances. Figure 18 shows an example C-shaped network of 100 nodes.

The graph found by G_2 is shown in Figure 19 which has far fewer edges than the original graph G (as shown in Figure 18). The graph in Figure 20 is G_3, which also has a large amount of savings. Figures 21 and 22 show $G_2 \cap G_3$ and G_{LMECN}, respectively. It is clearly evident that our proposed LMECN algorithm can produce topologies with significantly fewer edges even under irregular topologies.

4.4.2 Comparison of the Number of Links between G_2 and G_{LMECN}

We found the average number of links produced by LMECN under rigorous testing with various C-Shaped networks of 100, 150, 200 and 300 nodes. For each set of nodes we generated five different topologies. Nodes were deployed on a flat area of 1,000 m × 1,000 m. The com-

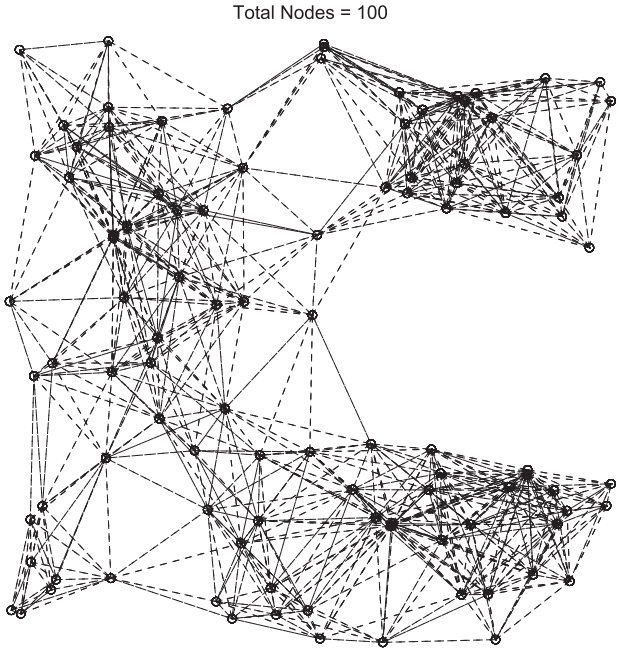

Figure 18. Original graph (C-shaped network) with 100 nodes

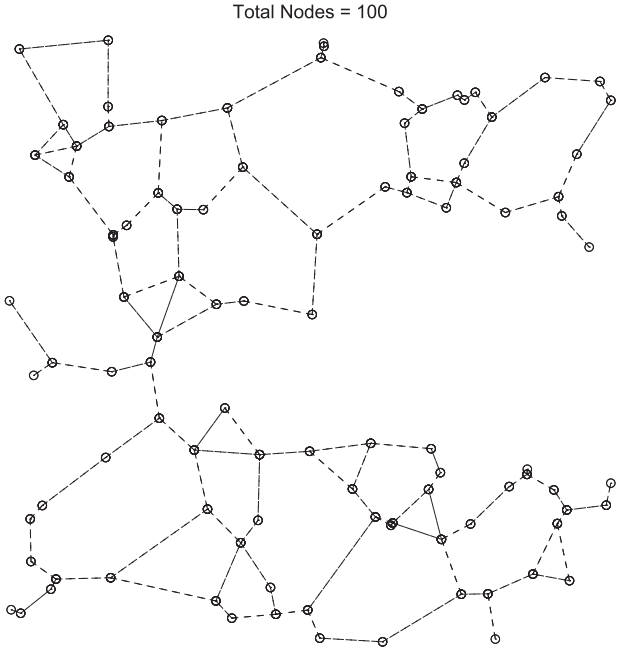

Figure 19. G_2 (C-shaped network) with 100 nodes

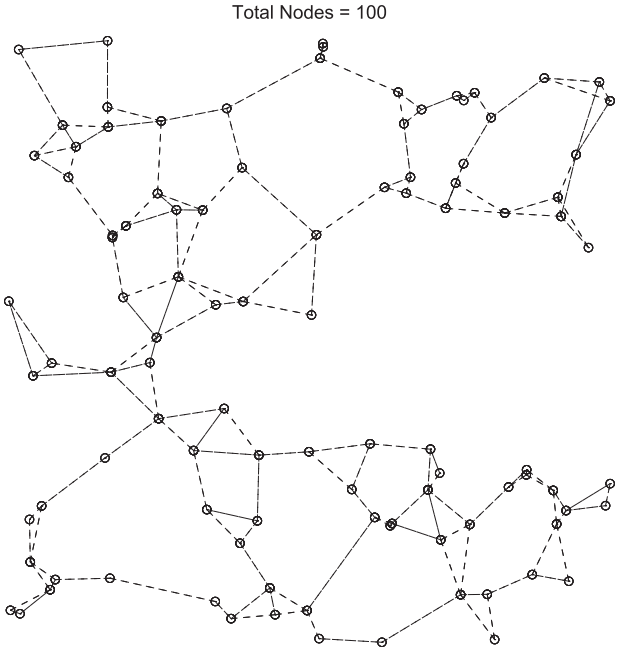

Figure 20. G_3 (C-shaped network) with 100 nodes

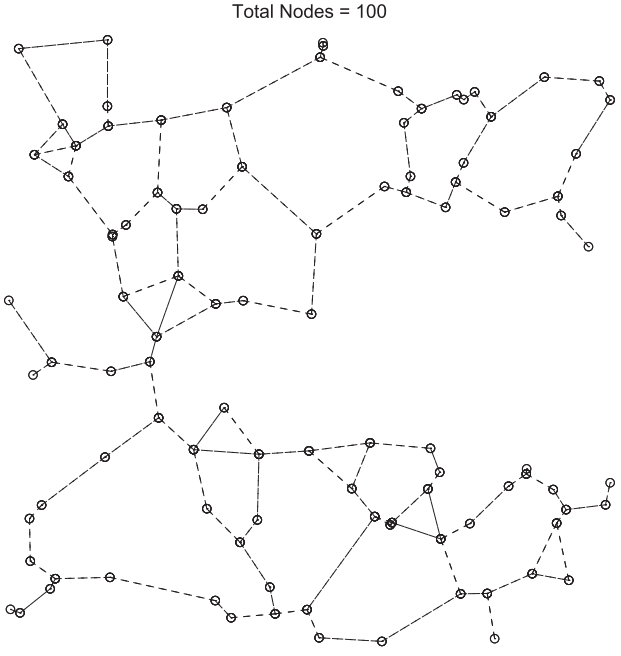

Figure 21. $G_2 \cap G_3$ (C-shaped network) with 100 nodes

munication range was set to 250 m. Nodes were carefully placed to form a shape of a 'C'. Figure 23 shows a comparison between the average number of edges in G_2 and in

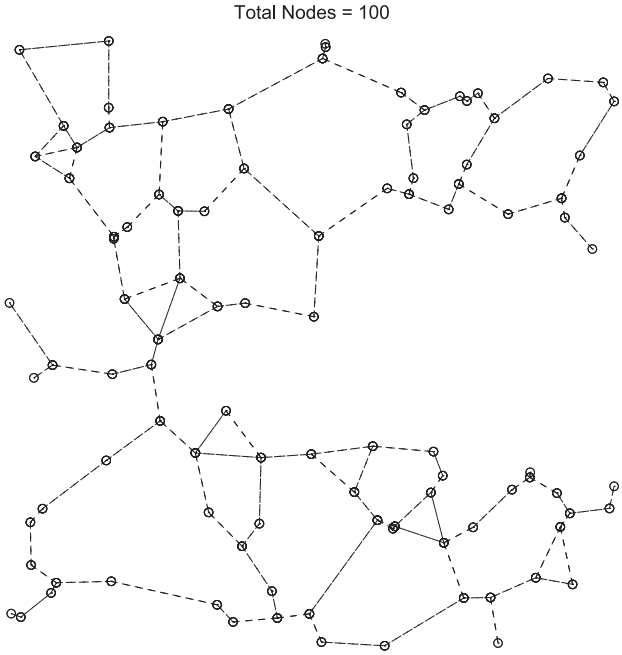

Figure 22. G_{LMECN} (C-shaped network) with 100 nodes

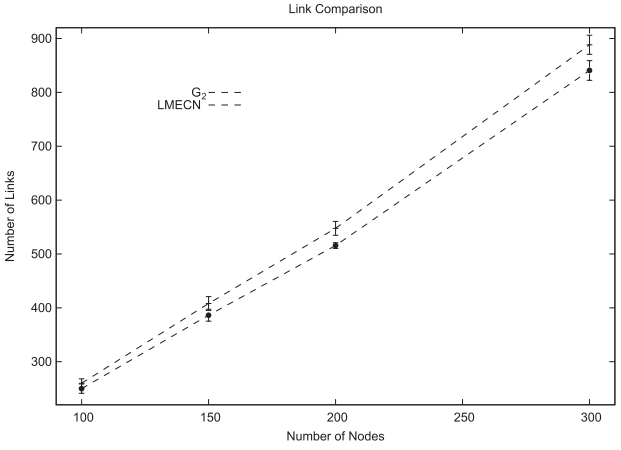

Figure 23. Comparison of the number of links between G_2 and G_{LMECN} under a C-shaped network

LMECN. Also 95% confidence intervals are shown with bars in the same figure. On the average, up to 5.81% of links are saved using LMECN algorithm.

4.4.3 Overhead Analysis

The overhead on a C-shaped network in LMECN again outperforms the construction of G_3. To analyze the overhead, we performed similar kinds of simulation used for

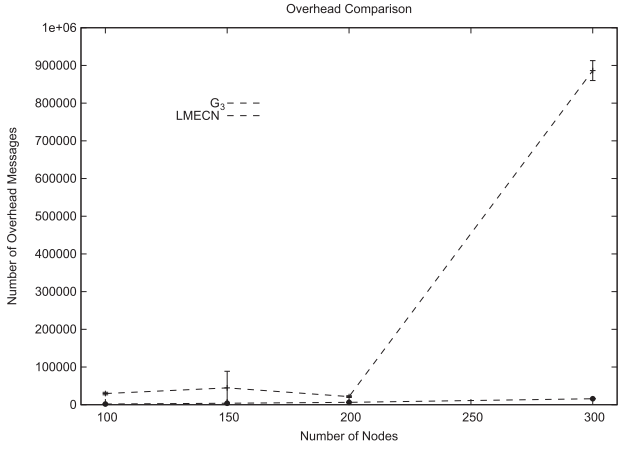

Figure 24. Overhead comparison between constructions of G_3 and G_{LMECN} under a C-shaped network

link analysis in the previous section. After running the simulation we took the average. Figure 24 shows the result with 95% confidence intervals using bars. LMECN incurs much less overhead than G_3 and savings are more significant in dense networks. As $G_2 \cap G_3$ requires the collection of two-hop information, therefore constructing such a graph requires a similar amount of overhead as constructing G_3.

5. Conclusion

The algorithm we have presented is a distributed algorithm that ensures minimum-energy path-preserving graphs in *ad-hoc* wireless networks. Our construction of LMECN is assisted by the rudimentary protocol of constructing G_2. Its need for information from an additional hop is served with no additional overhead and is obtained only through the previous information gained through G_2. Our studies, assisted by some simulation results, have demonstrated the superiority and robustness of the new algorithm over the previous solutions for networks with moderate and low density of nodes.

As a future work, we plan to investigate local distributed construction of some other higher-order subgraphs of $G_{\max}$ (i.e. G_4, G_5, etc.). We also plan to compare G_{LMECN} graph against the smallest minimum-energy path-preserving graph $G_{\min}$ to show how close it gets to the optimum.

6. References

[1] Rahman, A. and P. Gburzynski. 2006. MAC-assisted topology control for *ad-hoc* wireless networks. *International Journal of Communication Systems*, 19(9): 955–976.

[2] Li, L. and J. Halpern. 2001. Minimum energy mobile wireless networks revisited. *Proceedings of the IEEE International Conference on Communications (ICC)*, June 2001.

[3] Rahman, A. and P. Gburzynski. 2005. On constructing minimum-energy path-preserving graph for *ad-hoc* wireless networks. *Proceedings of International Conference on Communications ICC*, 2005.

[4] Ahmed, M., M. Shariar, S. Zerin and A. Rahman. 2008. Analysis of minimum-energy path preserving graphs for *ad hoc* wireless networks. *Proceedings of International Symposium on Performance Evaluation of Computer and Telecommunication Systems SPECTS*, 2008.

[5] Monks, J. P., V. Bhargavan and W. M. Hwu. A power controlled MAC protocol for wireless packet networks. *Proceedings of INFOCOM 2001*, pp. 219–228.

[6] Singh, S. and C. S. Raghavendra. 1998. Power efficient MAC protocol for multihop radio networks. *Proceedings of PIMRC*, 1998, pp. 153–157.

[7] Singh, S., M. Woo and C. S. Raghavendra. 1998. Power-aware routing in mobile *ad hoc* networks. *Proceedings of MobiCom*, 1998, pp. 181–190.

[8] Ramanathan, R. and R. Rosales-Hain. 2000. Topology control of multihop wireless networks using transmit power adjustment. *Proceedings of IEEE INFOCOM 2000*, Tel Aviv, Israel, March 2000, pp. 404–413.

[9] Narayanaswamy, S., V. Kawadia, R. S. Sreenivas and P. R. Kumar. 2002. Power control in *ad-hoc* networks: theory, architecture, algorithm and implementation of the COMPOW protocol. *Proceedings of the European Wireless Conference—Next Generation Wireless Networks: Technologies, Protocols, Services and Applications*, Florence, Italy, February 2002, pp. 156–162.

[10] Kawadia, V. and P. R. Kumar. 2003. Power control and clustering in *ad hoc* networks. *Proceedings of INFOCOM 2003*, San Francisco, CA, April 2003.

[11] Li, L., J. Y. Hlapern, P. Bahl, Y. Wang and R. Watenhofer. 2001. Analysis of a CON-based topology control algorithm for wireless multi-hop networks. *Proceedings of the ACM Symposium on Principle of Distributed Computing (PODC)*, 2001.

[12] Chen, Y.-S., S.-Y. Ni, Y.-C. Tseng and J.-P. Sheu. 1999. The broadcast storm problem in a mobile *ad hoc* network. *Proceedings of Mobicom*, 1999.

[13] Rodoplu, V. and T. Meng. 1999. Minimum energy mobile wireless networks. *IEEE Journal of Selected Areas in Communications*, 17: 1333–1344

[14] Li, N., J. C. Hou and L. Sha. 2003. Design and analysis of an MST-based topology control algorithm. *Proceedings of INFOCOM*, 2003.

[15] Rappaport, T. S. 1996. *Wireless Communications: Principles and Practice*, Prentice Hall, Englewood Cliffs, NJ, pp. 219–228.

[16] Niculescu, D. and B. Nath. 2001. *Ad hoc* Positioning System (APS). *Proceedings of IEEE Global Telecommunications Conference (GLOBECOM)*, Volume 5, pp. 2926–2931.

[17] Sang, Y. and W. Ruml. 2004. Improved MDS-based localization. *Proceedings of IEEE INFOCOM*, volume 4, March 2004, pp. 2640–2651.

Ashikur Rahman *received his BSc and MSc degrees in Computer Science and Engineering from the Department of Computer Science and Engineering, Bangladesh University of Engineering and Technology (BUET), Dhaka, Bangladesh in 1998 and 2001, respectively. He received his PhD in 2006 from the Department of Computing Science, University of Alberta, Canada. After finishing his PhD, he worked as a postdoctoral researcher at Simon Fraser University, Canada. His research interests include ad-hoc and sensor networks, peer-to-peer computing, swarm intelligence, back-end compiler optimization and neural networks. He is currently working as an Assistant Professor in the Department of Computer Science and Engineering, BUET.*

Mahmuda Ahmed *received her BSc degree in Computer Science and Engineering from the Department of Computer Science and Engineering, BUET, Dhaka, Bangladesh in 2008. She is now working as a Software Engineer in M&H Informatics (BD) Ltd (an IMS Health company). IMS Health is a global market research company working with pharmaceutical data. After 4 months of training in Switzerland regarding the local system and policy, she is currently working with the IMS Switzerland Team. Her research interest includes ad-hoc and sensor networks, distributed computing, data mining, query optimization, design of management information systems, database query languages, information retrieval, etc.*

Shobnom Zerin *has received her BSc degree in Computer Science and Engineering from the Department of Computer Science and Engineering, BUET, Dhaka, Bangladesh in January 2008. She is currently working as a software engineer at M&H Informatics (BD) Ltd (an IMS Health company). Her research interests include ad-hoc and sensor networks, optical and wireless communication, and network controlled systems.*