INTERNATIONAL JOURNAL OF COMMUNICATION SYSTEMS
Int. J. Commun. Syst. 2005; **00**:1–6 *Prepared using* dacauth.cls *[Version: 2002/09/18 v1.01]*

MAC-assisted topology control for ad-hoc wireless networks

A. Rahman and P. Gburzynski*

University of Alberta, Department of Computing Science, Edmonton, Alberta, CANADA, T6G 2E8

SUMMARY

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. We propose a new algorithm for constructing minimum-energy path-preserving subgraphs of G_{max}, i.e., ones minimizing the energy consumption between node pairs. Our algorithm involves a modification to the Medium Access Control (MAC) layer. Its superiority over previous solutions, up to 60% improvement in sparse networks, demonstrates once again that strict protocol layering in wireless networks tends to be detrimental to performance. Copyright © 2005 John Wiley & Sons, Ltd.

KEY WORDS: ad-hoc networks; topology control; connectivity; power constraints; connected graphs; medium access control

1. INTRODUCTION

Energy consumption control is one of the key design issues in ad-hoc wireless networks, with transmission power being the predominant factor in the overall energy budget. One natural formulation of the power control problem is choosing the minimum power level by each node, based on the position information of the reachable nodes, while maintaining global connectivity. Such a model assumes that the minimum power needed to reach a node depends solely on the distance to the node. This assumption implies the symmetry of the problem with respect to the two endpoints of a transmission path.

1.1. Motivation

Consider 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 $G = (V, E)$, with the vertex set

*Correspondence to: University of Alberta, Department of Computing Science, Edmonton, Alberta, CANADA, T6G 2E8

Received 10 March 2005
Accepted 3 October 2005

V representing the nodes, and the edge set defined as follows:

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

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 defined this way is called the *maximum powered network*.

The local choices regarding the transmission power at individual nodes will collectively shape a subgraph of the maximum powered network. The properties of that subgraph, e.g., its average degree, may have a significant impact on the performance of the routing layer. For example, flooding, used as a typical way of route discovery, may cause serious *broadcast storm* problems [20] in a dense graph, e.g., one close to the maximum powered network. By reducing the power level at each node we also reduce the average node degree, which, in turn, reduces the contention in the node's neighborhood. Thus, it is generally beneficial to be able to broadcast route discovery messages over a proper subgraph of G.

As pointed out in [7], efficient and effective power control has other numerous advantages. In a nutshell, the transmitted power level affects the throughput capacity of the entire network. It also determines the degree of congestion affecting the performance of the transport layer. Needless to say, it also directly relates to the magnitude of interference at the receivers located nearby the sender.

1.2. The problem statement

The issue of selecting the optimum transmission power formulated in this context was first tackled by Roduplu et al. [17] who considered the so-called *enclosure graphs*. The local enclosure graphs constructed for individual nodes form globally a strongly connected graph guaranteed to contain the minimum-energy paths for all pairs of nodes. By applying to that graph a distributed Bellman-Ford algorithm with energy as the cost metric, one can find a minimum-energy end-to-end path for any pair of nodes.

We say that a graph $G' \subseteq G$ 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, at least one of the (possibly multiple) minimum energy paths between u and v in G also belongs to G'. Minimum-energy path-preserving graphs were were first defined in [8]. Typically, many minimum-energy path preserving graphs can be formed from the original graph G. It has been shown that the smallest of such subgraphs of G is the graph $G_{min} = (V, E_{min})$, where $(u, v) \in E_{min}$ iff there is no path of length greater than 1 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 $G = (V, E)$ such that $(u, v) \in E_i$ iff $(u, v) \in E$ and there is no path of length i that requires less energy than the direct one-hop transmission between u and v. Then G_{min} can be formally defined as follows:

$$G_{min} = \bigcap_{i=2}^{n-1} G_i$$

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

Distributed construction of G_{min} by the nodes is somewhat tricky, and it contradicts its own goals, because it requires communicating with distant nodes using high power. On the other hand, graphs G_i can be built based on local information at considerably relaxed power requirements. An algorithm for constructing one such graph, G_2, was presented in [8]. It works reasonably well in dense networks but its performance degrades considerably when the network density drops to medium or low.

Note that while G_2 is only the first of the series G_i, it is the most interesting and most practically useful derivative of the maximum powered graph G. By increasing the value of i we try to account for longer and longer paths (with higher number of hops) that may turn out to require less energy than a direct hop. Because of the obvious facts that 1) the number of hops tends to directly correlate to distance, 2) the total transmission power of a path is additive on the number of hops, the likelihood of such paths drops rapidly once the case $i = 2$ has been handled. Consequently, considering that the cost of discriminating among longer paths will unavoidably involve exchanging many messages, and thus will incur obvious energy overheads, it makes perfect sense to restrict our attention to G_2.

In this paper, we show how to construct G_2 efficiently in sparse and moderately dense networks with some assistance of the Medium Access Control (MAC) layer. The proposed modifications affect the backoff procedure of the 802.11 collision avoidance scheme and are somewhat reminiscent of the previous work [3] on Quality of Service issues related to fairness and priority scheduling. In our own previous work [14], we proposed another modification to the collision avoidance mechanism of 802.11 aimed at improving the reliability of multicast transmissions in ad-hoc networks.

2. RELATED WORK

A significant amount of research has been directed at power control algorithms for wireless mobile networks. One can see three generic approaches. The first class of solutions looks at the issue from the perspective of MAC layer. In particular, Monks *et al.* [12] propose a modification of IEEE 802.11's RTS-CTS handshake procedure. They argue that a node, say w, overhearing the handshake between nodes u and v can estimate its distance to the recipient, say v, from the strength of v's CTS packet. Then, w can continue transmitting at an appropriately tailored power level in a way that will not interfere with u's reception at v. Sing *et al.* [18] propose powering-off transceivers—to reduce energy consumption—when they are not actively transmitting or receiving any packets.

The second approach, dealing with the network layer, can be dubbed *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. Some of those metrics, mentioned in [19], are: energy consumed per packet, time to network partition, variance in node power levels, and (power) cost per packet.

The third approach strives to separate routing from topology control, although, power-aware routing protocols are often used in conjunction with solutions in this class. The proposed algorithms address different goals subject to different constraints. Our focus is on those algorithms that are distributed, lead to power efficient operation, and preserve network connectivity. We shall now discuss some of them in brief.

Ramanathan et al. [15] describe two centralized algorithms to minimize the maximum

Prepared using dacauth.cls

transmission power used by any node while maintaining connectivity or bi-connectivity. Two distributed heuristics called *Local Information No Topology* (LINT) and *Local Information Link-State Topology* (LILT) are introduced to deal with the dynamics of the mobile environment. In LINT, each node is characterized by three parameters related to its degree: the desired node degree d_d, the upper limit on node degree, d_h and the lower limit on node degree d_l. Every node periodically checks its dynamic degree constructed from the neighbor table and adjusts the transmission power to keep the degree within the threshold limits. LILT is an improvement over LINT which overrides the high limit on node degree, if topology changes cause undesirable connectivity patterns. Neither heuristic absolutely preserves connectivity, even if it is achievable in principle, i.e., the maximum powered graph G is connected.

Cone-Based Topology Control (CBTC), proposed by Li *et al.* [9], generates a graph structure similar to the one proposed by Yao in [24]. 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 $\alpha \leq \frac{5\pi}{6}$, the resultant graph is connected, provided the original one (G) 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 (see Section 5).

Narayanaswamy *et al.* [13] present a power control protocol named COMPOW. Their goal is to choose the smallest common power level by each node that 1) preserves connectivity, 2) maximizes traffic carrying capacity, 3) reduces contention in the MAC layer and 4) requires low power to route packets. In their approach, several routing daemons run in parallel at each node, one for each (discretized) power level. Each routing daemon exchanges control messages with its counterparts at the neighboring nodes (at the specified power level) to maintain its own routing table. The entries in different routing tables are compared to determine the smallest common power that ensures the maximum number of connected nodes. One serious flaw of this approach is its assumption of the uniform distribution of nodes, which is impractical. If nodes are deployed in a non-homogeneous fashion, then, for example, a single node located some distance apart from a group of close nodes may significantly affect the performance of the entire group.

CLUSTERPOW [6] was designed to overcome some of the shortcomings of COMPOW by accounting for non-uniform distributions of nodes. It introduces a hierarchy, whereby closely located nodes are allowed to form a cluster and choose a small common power to interact with each other. Different clusters communicate among themselves at a different (higher) power level. Intentionally, most of the intra-cluster communication is done at a lower power level, and the (possibly rare) inter-cluster communication is carried out with a higher power. As before, each node runs multiple daemons, which constantly exchange reachability information with neighbors. This incurs a significant message overhead.

N. Li *et al.* [10] 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 (≤ 6) 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.

Rodoplu *et al.* [17] introduce the notion of *relay region* based on a specific power model. Their algorithm was later modified by Li *et al.* [8] to trim some unnecessary edges. Our work closely relates to these two studies, and a detailed discussion of these algorithms will be given in the next section.

For other related work in this area, Wattenhofer *et al.* [23] describe a two-phased algorithm, which consists of creating a variation of the Yao graph followed by a Gabriel Graph. The combined structure of Yao graph and Gabriel graph has been shown to be more sparse and still offer a constant bound on the *energy stretch factor.* Also, Huang et al. [5] propose a topology control algorithm taking advantage of directional antennas.

Recent research has also shown a tremendous interest in topology control as a means of interference reduction. Bukhart *et al.* [2] contradict the natural presumption that the sparseness of topology implies low interference. They provide an intuitive definition of interference and, based on that definition, propose algorithms to construct connected subgraphs and spanners with minimum interference. Unfortunately, their solution does not preserve the minimum-energy paths between node pairs. Another algorithm, with the same flaw, is presented in [11], where the problem of minimizing the average or maximum interference (per link or node) is studied. Tang *et al.* [21] propose an algorithm for interference-aware topology control in multi-channel mesh networks based on IEEE 802.11. They provide a novel definition of co-channel interference and present efficient heuristics for channel assignment to the network such that the induced topology is interference-minimal. These issues, however, are beyond the scope of our paper, which focuses on single-channel environments.

3. MINIMUM-ENERGY PATH PRESERVING GRAPHS

3.1. Power model

We assume the well known, generic, two-ray, channel path loss model, where the minimum transmission power is a function of distance [16]. To send a packet from node x to node y, separated by distance $d(x, y)$, the minimum necessary transmission power is approximated by

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

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

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

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

3.2. Previous approach to constructing G_2

The algorithm presented in [17] is based on the notion of *relay region.* Throughout the paper we will refer to that algorithm as R&M. 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(a)).

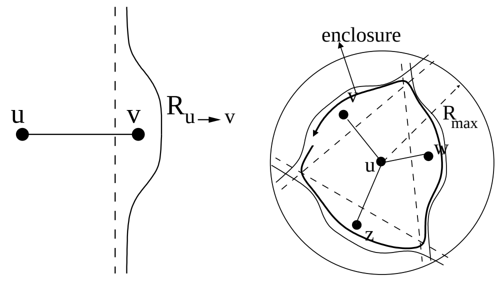


Figure 1. (a) The relay region of v with respect to u, (b) The enclosure of node u

Given the definition of relay region, the algorithm for constructing G_2 becomes straightforward. Suppose that u is the starting node of a path. If, as perceived by u, some node w falls in the relay region of some other node v, then w will not be included in the so-called *neighborset* of u (i.e., u will not transmit directly to w). Thus, by definition, the neighborset of node u will contain only those nodes that do not fall into relay regions of other nodes reachable by u. G_2 can be constructed by connecting each node with only those nodes that are included in its neighborset.

The efficiency of constructing G_2 using this approach depends on how inexpensively nodes can collect the position information of their neighbors. One trivial way to discover the position of all nodes in the neighborhood is to periodically broadcast a *neighbor discovery message* (NDM) at the maximum power P_{max}, to which all reachable nodes will respond with their position information.

With power concerns in mind, it is natural to ask this question: "Is there a way for node u to restrict the search area to a subset of its transmission range?" Perhaps, in sufficiently many cases, u does not require the position information of all nodes that fall within its communication range to determine its neighborset. As it turns out, such confinement is often possible.

Given $R_{u \to v}$, the relay region of node v with respect to node u, the complement of this region, denoted by $R^c_{u \to v}$, is the set of points for which it is not power-efficient for u to use node v as a relay. Let $N(u)$ be the set of nodes that do not fall in the relay region of any other node in u's neighborhood. Then, $\bigcap_{k \in N(u)} R^c_{u \to k}$ is the set of points where u should transmit directly without using any relay. On the other hand, the direct transmission range of u is limited by P_{max}—the maximum transmission power. Let $F(u, P_{max})$ denote the circular region with radius R_{max} centered at u and describing its transmission range. The *enclosure* of node u is defined as the following set of points:

$$\epsilon_u = \bigcap_{k \in N(u)} R^c_{u \to k} \bigcap F(u, P_{max})$$

Figure 1(b) shows an example of enclosure. It is its enclosure beyond which a node need not search for neighbors. This observation lead in [8] to a power saving algorithm for constructing

G_2. In a nutshell, instead of broadcasting the NDM at the maximum power, u will start with some initial power, $P_0 \ll P_{max}$. After collecting responses from the neighborhood, if the enclosure has been found, then there is no need to search any further. Otherwise u will re-broadcast the NDM at an increased power level and try again. This process will continue until u either finds the enclosure or reaches P_{max}. Figure 2 gives a high level description of that algorithm, which we shall refer to as the *Reduced Neighbor Search Algorithm*, or RNSA for short. Note that the efficiency of RNSA depends on the number of iterations required to find the enclosure, which in turn depends on the initial power P_0 and the power increment P_{inc} applied in step 6.

Algorithm RNSA:
1. *transmission power* := P_0
2. loop
3. Broadcast NDM and collect responses.
4. Update the neighborset using the definition of relay region.
5. If enclosure found or *transmission power* = P_{max} then exit.
6. Increase *transmission power* by P_{inc}.
7. endloop

Figure 2. Reduced Neighbor Search Algorithm (RNSA)

3.3. Problems with RNSA

RNSA suffers from two major drawbacks. First, while the algorithm works fine when the network is dense, in a sparse or moderately populated network it tends to exhibit poor performance. To see the reason for this, let us note that the enclosure of a node u can be formed in one of two possible ways:
Case (i): the enclosure is determined solely by the nodes in $N(u)$. This happens when the following condition holds:

$$\bigcap_{k \in N(u)} R^c_{u \to k} \subseteq F(u, P_{max})$$

Such an enclosure is the intersection of the complements of the relay regions of all nodes in $N(u)$ (see Figure 1(b)). We will call it an *enclosure by neighbors.*
Case (ii): the transmission range of u is the limiting factor, i.e.,

$$\bigcap_{k \in N(u)} R^c_{u \to k} \neq \bigcap_{k \in N(u)} R^c_{u \to k} \bigcap F(u, P_{max})$$

Such enclosures are called *enclosures by maximum boundary* (see Figure 3).

If a node has an enclosure by neighbors, then, in principle, it need not transmit at P_{max} to find that enclosure. For such nodes, RNSA is useful and may bring about power savings compared to the naive scheme. On the other hand, a node having an enclosure by maximum boundary, will ultimately need to search with maximum power. For such a node, RNSA performs worse than the naive scheme as it runs through a number of essentially futile iterations

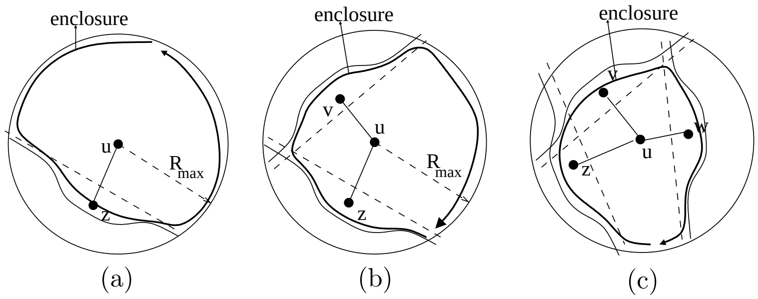


Figure 3. Enclosures by maximum boundary

before reaching P_{max}. The traffic caused by the NDMs broadcast during those iterations and the multiple repetitive responses to those messages wastes bandwidth and contributes to the noise in the neighborhood.

One can naturally expect that the likelihood of finding a node whose enclosure is determined by neighbors is higher in a dense network and at locations further from the network's edge. On the other hand, sparse networks will have many nodes with maximum boundary enclosures.

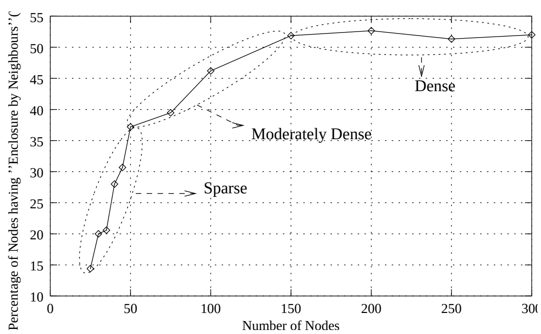
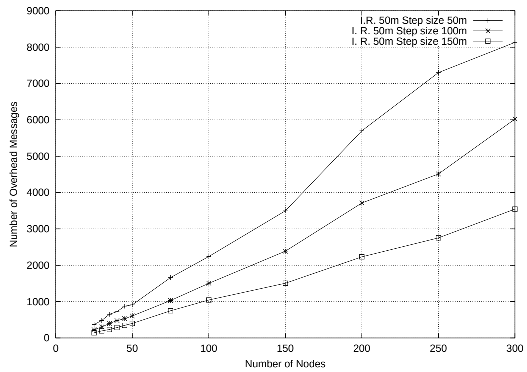


Figure 4. Percentage of nodes with enclosures by neighbors

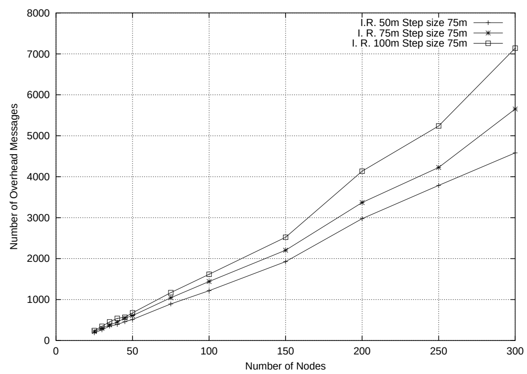
Figure 4 shows some statistics relating the observed percentage of nodes with enclosures by neighbors to the network density. The density of the network in this experiment was determined by the total number of nodes, which were distributed uniformly in a fixed square region of $670m \times 670m$. Each point was obtained as the average of 5 distribution samples.

The percentage of nodes having enclosures by neighbors is between 14% and 38% when the total number of nodes is less than 50 (sparse network), between 38% and 52% for $50 - 150$ nodes (moderately dense network), and greater than 52% for the total number of nodes greater than 150 (dense network). This picture clearly suggests that RNSA will not perform well for sparse or moderately dense networks, where most nodes have to transmit at P_{max} to find their enclosures.

The second problem with RNSA is the lack of clear guidelines regarding the selection of the initial power P_0 and the increment P_{inc}. Figure 5, showing the relationship between those parameters and the resulting message overhead of RNSA, demonstrates that their choice is not irrelevant.



(a)



(b)

Figure 5. Comparing number of overhead messages by varying (a) step size, (b) initial communication range

For simplicity, the power level shown in Figure 5 has been transformed into the transmission range (see Section 3.1). In part (a), the initial transmission range is the same for all three curves ($50m$), but the increments are different: $50m$, $100m$, and $150m$. Especially for dense networks, where RNSA is most useful, the differences are considerable and exceed 50%.

In part (b), the step size is fixed at $75m$, while the initial transmission range varies between $50m$ and $100m$. Again, the selection of P_0 appears to affect the observed overhead quite significantly.

Of course, the simple exercise illustrated in Figure 5 does not allow us to draw quantitative conclusions regarding the recommended setting of the two parameters of RNSA. As the observed susceptibility of the algorithm's performance to those parameters is rather high, one can expect that their optimum setting is also highly sensitive to various characteristics of the network. As those characteristics in ad-hoc networks tend to be diverse and often dynamic, there is little hope that the algorithm can dynamically adapt itself to offer its best performance in every possible configuration.

4. CONSTRUCTING G_2 FOR SPARSE AND MODERATELY DENSE NETWORKS

In this section we describe an algorithm for constructing G_2 that works more efficiently than RNSA in sparse and moderately dense networks. Our algorithm is named BICOMP, for *BIased COntention at Maximum Power*. We shall start by defining some terms.

4.1. Cover region and cover set

Consider a pair of nodes (s, f), such that f lies within the communication range of s, i.e., is reachable by s at P_{max}. Envision the set of all points that can possibly act as relays between s and f, such that it would be more power efficient for s to use an intermediate node located at one of those points instead of sending directly to f. Note that, owing to the symmetry of our underlying propagation model, exactly the same set is defined by considering f as the starting point. We will call it the *cover region* of s and f and denote by $C_{(s,f)}$. The collection of all nodes falling into the cover region of s and f will be called the cover set of s and f. Formally the cover region and cover set, are described by the following definition.

Definition 1: The cover region $C_{(s,f)}$ of a pair of nodes (s, f), where f is reachable from s, is defined as:

$$C_{(s,f)} = \{<x,y> \mid td^{\alpha}(s,<x,y>) + td^{\alpha}(<x,y>,f) + r \leq td^{\alpha}(s,f)\} ,$$

where $\alpha \geq 2$. In the above equation, $d(s,<x,y>)$ denotes the distance between node s, and a hypothetical node located at $<x,y>$, and r is the fixed receiving power. The cover set of the same pair (s, f) is

$$\xi_{(s,f)} = \{v \mid v \in V \wedge Loc(v) \in C_{(s,f)}\}$$

Figure 6 shows two examples of cover regions, with the path loss exponent $\alpha = 2$ and $r = 0mW$, and $\alpha = 4$, $r = 20mW$.

Lemma 1 : (a) For any $c \in \xi_{(s,f)}$, $d_{sc} < d_{sf}$, (b) If $c \in \xi_{(s,f)}$ then $f \notin \xi_{(s,c)}$.

Proof : (a) If $c \in \xi_{(s,f)}$ then from definition 1 it follows that, $d^{\alpha}_{sc} + d^{\alpha}_{cf} + r/t \leq d^{\alpha}_{sf}$. Now, for $r > 0$ and $\alpha \geq 1$, $d_{sf} > d_{sc}$.
(b) If $c \in \xi_{(s,f)}$, then from (a) $d_{sf} > d_{sc}$. Now suppose that also $f \in \xi_{(s,c)}$ then again from (a), $d_{sc} > d_{sf}$ which is a contradiction.

Note the difference between relay regions and cover regions. Given a node pair (u, v), the relay region provides an answer to this question: "What are the points for which node v can act as a power-efficient relay for node u?" On the other hand, the question answered by the cover region is: "What are the points that can act as power-efficient relays for node u when sending to v?" These questions are quite different; in particular, cover regions are indifferent to the ordering of u and v, while relay regions are not.

4.2. Constructing G_2

As in RNSA, the operation is described from the viewpoint of one node s. In contrast to RNSA, s broadcasts a single neighbor discovery message (NDM) at the maximum power P_{max}. For

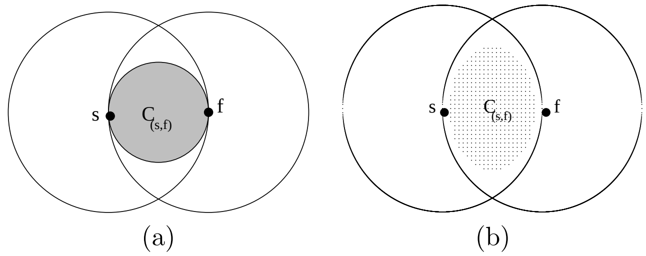


Figure 6. Cover regions: (a) $\alpha = 2$, $r = 0mW$ (b) $\alpha = 4$, $r = 20mW$

now, let us assume that all nodes receiving the NDM from s send back a reply. Later we will explain how the number of replies can be reduced with the assistance of the MAC layer. The reduced overhead in our algorithm will result from the reduced number of replies in a dense network. As it turns out, those savings outweigh the gains of the reduced power transmission of the NDM in RNSA, especially in networks that are not very dense.

While s collects the replies of its neighbors, it learns their identities and locations. It also constructs the cover sets of those neighbors. Initially, all those sets are empty (s doesn't even know what neighbors there are). The set A_s, which also starts empty, keeps track of all the nodes discovered in the neighborhood.

Whenever s receives a reply to its NDM from a node v, it performs the algorithm listed in Figure 7. Its purpose is to update the configuration of the cover sets. At the end, when s has received all the replies, the configuration of cover sets is complete.

updateCoverRegion(s, v)
begin
  $for\ each\ w \in A_s$
    $if\ Loc(v) \in C_{(s,w)}\ then$
      $\xi_{(s,w)} = \xi_{(s,w)} \cup \{v\};$
    $else\ if\ Loc(w) \in C_{(s,v)}\ then$
      $\xi_{(s,v)} = \xi_{(s,v)} \cup \{w\};$
  $A_s = A_s \cup \{v\};$
end

Figure 7. Algorithm for building cover sets

The goal of node s is to determine its *neighborset*, i.e., the collection of neighbors to which transmission should be direct. Having determined the cover regions of all its neighbors, s is in position to identify the members of its neighborset. If $\xi_{(s,v)} \neq \emptyset$ for some v, it means that sending directly to node v is not power efficient: there is at least one node $w \in \xi_{(s,v)}$ that can act as a power-efficient relay between s and v. On the other hand, a node v that has no nonempty cover set with s, but belongs to the neighborhood of s, i.e., is present in A_s, necessarily has no power-efficient relays and thus belongs to the neighborset of s. Consequently, the loop listed in Figure 8 completes the algorithm by generating the neighborset of s denoted by $\aleph_s$.

neighbor(s)
begin
  $\aleph_s = \emptyset$
  for each $v \in A_s$
    if $\xi_{(s,v)} = \emptyset$ *then*
      $\aleph_s = \aleph_s \cup \{v\}$;
end

Figure 8. Generating the neighborset of s

4.3. BICOMP: reducing the number of reply messages

The primary advantage of BICOMP is that it is able to reduce the number of reply messages, and thus significantly lower the overall power expense needed to discover the resultant graph. Consider a simple scenario where s can reach only two nodes v and w within the radius of maximum transmission range, such that $v \in \xi_{(s,w)}$. Clearly, from Lemma 1(b), $w \notin \xi_{(s,v)}$. The neighborset of s, computed by the algorithm in Figures 7 and 8, $\aleph_s = \{v\}$. When node s broadcasts its neighbor discovery message, nodes v and w are supposed to send back a reply message with their location information.

Both nodes v and w, will contend for access to the shared wireless channel to send their reply messages back to s. If v wins, then the reply of w will be received by s after the message sent by v. Note that the message sent by w will be redundant: it will not affect the outcome of the algorithm, as w is covered by v and it should not be included in $\aleph_u$. On the other hand, if w wins and sends its reply first, the algorithm will first add w to A_u and then, after receiving the second message from v, add v to $\xi_{(s,w)}$.

Note that if v were allowed to win, and w overheard the reply of v, then w could refrain from sending its reply to s. Node w is in the same position as s to find out that its message is redundant: it has the location information of node s (which arrived in the NDM of s) and can carry out exactly the same simple calculations as node s. This way, some replies can be eliminated before being transmitted.

In a general scenario, we would like to be able to enforce some ordering of the reply messages that would give precedence to those likely to be relevant and postpone those likely to be redundant. A node detecting that its pending message is redundant would drop it and thus reduce the amount of traffic needed for neighborset discovery.

To be able to order the reply messages, we need to exercise some control over the contention resolution mechanism used in the MAC layer. With IEEE 802.11, a node willing to transmit a packet under contention has to wait for a certain number of idle slots chosen at random in the range of $[0, cw-1]$, where cw is the so-called *contention window*. Statistically, different nodes are likely to pick different numbers, which will help them transmit without interference in different time slots. To influence the order of transmissions in a way that would be compatible with our sense of relevance of the reply messages, we have to bias the random distribution of the slot selection process.

Note that generally we cannot eliminate randomness from the process. Whatever idea a node may have regarding the selection of it transmission slot, the decision is always local and thus cannot preclude other nodes from arriving at the same decision. This may happen when two or more nodes find themselves in the same (or similar) situation with respect to s and conclude that their priorities are the same. To avoid permanent lockouts in such situations, the contention resolution scheme must not give up its random component.

Our intention is to make the expected waiting time (the number of skipped slots) an increasing function of the distance from the node that sent the NDM. This will increase the chance that covered nodes will schedule their transmissions later and, consequently, the chance that those transmissions will never take place. According to Lemma 1(a), if a node v is in the cover set of node w, then d_{sv} must be less then d_{sw}. A natural way to proceed is to divide the area around node s into partitions according to the distance from s.

4.3.1. Equal-area partitions Let $F(s, P_{max})$ represent the circular region of radius R_{max} reachable by s at its maximum transmission power. We divide $F(s, P_{max})$ into n equal-area partitions. A node v is said to fall into partition i, $1 \leq i \leq n$ iff,

$$\begin{cases} 0 < d_{sv} \leq R_{max} \times \sqrt{\frac{i}{n}} & \text{when } i = 1 \\ R_{max} \times \sqrt{\frac{i-1}{n}} < d_{sv} \leq R_{max} \times \sqrt{\frac{i}{n}} & \text{when } i = 2 \dots n \end{cases}$$

Note that, in this scheme, the circle $F(s, P_{max})$ centered at node s is divided into n partitions, all with the same area of

$$A = \frac{\pi R_{max}^2}{n}$$

One can argue that partitioning nodes this way makes sense because, assuming the uniform distribution of nodes, every partition will tend to contain about the same number of nodes. The issue is far from being that simple, however. This is because nodes located closer to s are more likely to participate in the neighborset. Consequently, it may be sensible to group more distant nodes into larger classes, providing for finer contention resolution in a closer neighborhood of s.

4.3.2. Equal-width partitions With this scheme, $F(s, P_{max})$ is divided into n equal-width partitions. A node v is said to fall into partition i, $1 \leq i \leq n$ iff,

$$\frac{R_{max} \times (i-1)}{n} < d_{sv} \leq \frac{R_{max} \times i}{n}$$

This time, the circle $F(s, P_{max})$ centered at node s is divided into n circular strips with the same width of R_{max}/n. The area of partition i is

$$A_i = \frac{\pi R_{max}^2 (2i-1)}{n^2}$$

and it increases with the partition radius.

4.3.3. Modifying the backoff mechanism A node falling into partition i chooses a random number prescribed by the following formula:

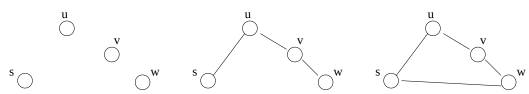
$$R = (i-1) \times \frac{cw}{2^{\lceil log_2 n \rceil}} + \lfloor U(0,1) \times \frac{cw}{2^{\lceil log_2 n \rceil}} \rfloor \ ,$$

where $U(0, 1)$ is a uniform distribution between 0 and 1. If n is a power of 2, the equation becomes a bit simpler.

Example: Let $n = 2$. The transmission range of s is divided into two partitions. According to the equal-width scheme, the nodes whose distance from s is less than $R_{max}/2$ are be assigned to partition number 1, and all the remaining nodes fall into partition 2. Assuming the contention window size $cw = 32$, the nodes in partition 1 choose a random number between 0 and 15 and the nodes in partition 2 select a random number between 16 and 31.

4.4. Extraneous edges

Consider the simple scenario shown in Figure 9(a). Suppose s is sending an NDM message. In this configuration, $u \in \xi_{(s,v)}$ (and also $v \in \xi_{(s,w)}$), but $u \notin \xi_{(s,w)}$. Both RNSA and R&M do not depend on the ordering of reply messages. The final graph produced by RNSA and R&M is shown in Figure 9(b) and (c) respectively. Note that the graph constructed by R&M has one edge more than the one produced by RNSA. The shape of the final graph found by our algorithm depends on the order of reply messages. If u sends its reply before v, then v will cancel its reply because it is covered by u. Later on, when w sends its reply, s will not be able to detect that w is covered by v (s does not know v's position because v has canceled its message) and will add an extra edge between s and w, similar to R&M.



(a) (b) (c)

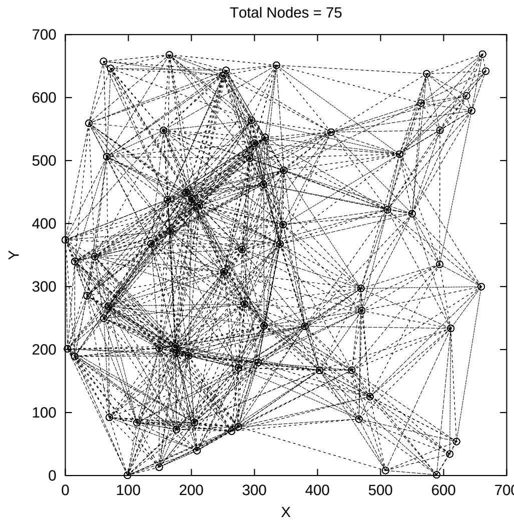
Figure 9. An extraneous edge

Now, if v sends its reply before u, then the edge between s and w will not show up in the final graph (because s will know the position of v). Thus, in that case, our algorithm will produce the same graph as RNSA. In other words, the exact outcome of our algorithm depends to a certain extent on the ordering of reply messages. In the best case, the algorithm will produce exactly G_2, in the worst case it will generate a graph similar to the output of R&M, and on the average it will produce something in between.

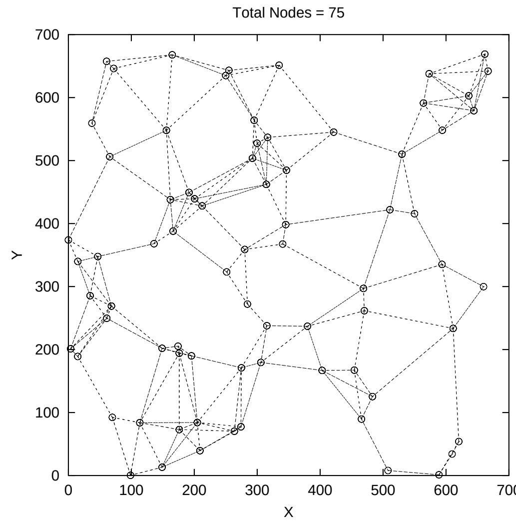
Note that there is a relatively easy way to modify BICOMP to avoid inserting the extraneous edges, and to produce exactly G_2. For that, we need to be able to reschedule some of the previously canceled messages. For example, in the scenario shown in Figure 9, having canceled the reply message and then overhearing the reply sent by w, node v may reschedule its reply to notify s about its location. Of course, this will result in an increased number of reply messages (and thus a higher power overhead), especially in scenarios more complex than the one illustrated in Figure 9.

On the other hand, the likelihood of the extraneous edges diminishes in situations where the problematic node is covered by multiple nodes (as is likely in many practical scenarios). Although one of those covering nodes may cancel its reply, there exist other covering nodes whose replies may make it to s, which will then be able to eliminate the extra edge. The probability that all covering nodes will cancel their replies may turn out to be sufficiently low to be acceptable.

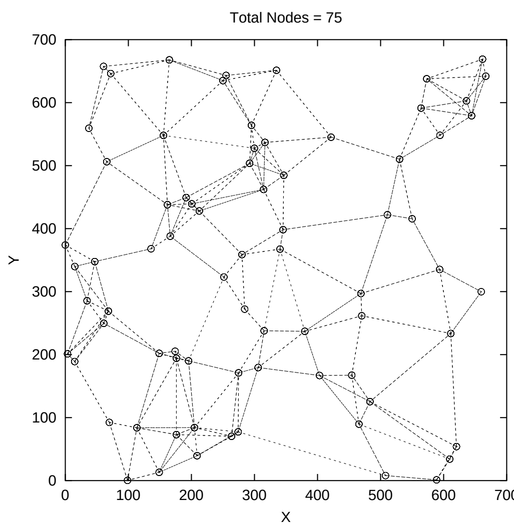
We have carried out experiments to assess the magnitude of the problem, i.e., find out how many extraneous edges tend to be included by BICOMP. The results show that the percentage



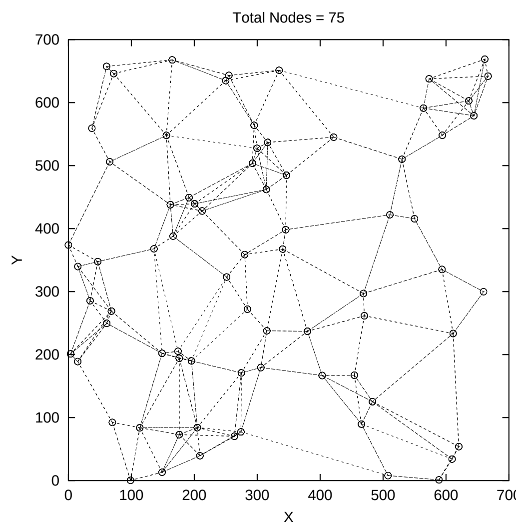
(a) Original graph



(b) G_2 generated by RNSA



(c) LOHG



(d) Generated by R&M

Figure 10. Subgraphs obtained by different algorithms

TABLE I
RESULTANT GRAPH SIZE

Number of nodes	Edges in G_2	Edges LOHG	Edges in R&M
25	68	68	70
30	78	80	82
35	104	110	112
40	142	148	148
45	182	188	192
50	184	188	192
75	352	362	368
100	622	634	638
150	1244	1248	1248

of those edges is quite low. For illustration, Table I shows a comparison between the number of edges in G_2, those constructed by BICOMP (as described in Section 4.3), and those found by R&M. Note that, owing to the random character of contention resolution, the numbers for BICOMP reflect one of several possible outcomes, which are always bound from above by the last column. The possible reduction in the number of edges—acquired by complicating BICOMP to reschedule some of the dropped replies—appear to be insignificant, and they do not warrant the added power expense. Consequently, we have decided to ignore the issue and not to reschedule any canceled reply messages. The final topology produced by our algorithm, which may be slightly larger than G_2, will be called a *Low OverHead Graph*, or LOHG for short. One can easily see that the following Lemma holds.

Lemma 2 : (a) LOHG contains G_2. (b) LOHG is connected.

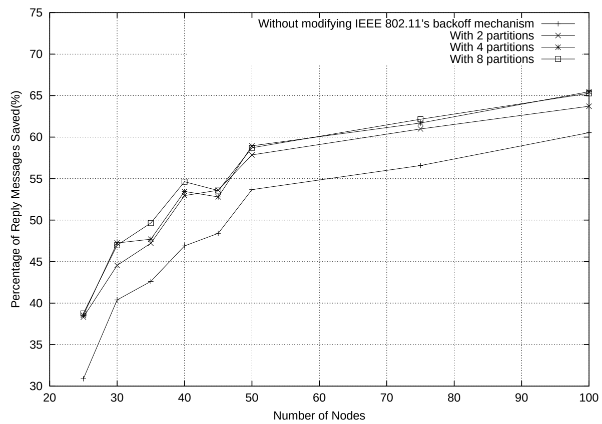
Proof : (a) According to our algorithm each node u starts the process by broadcasting an NDM message. Suppose the set of nodes receiving this NDM message is $N(u)$. Each node v receiving the NDM message will initially schedule a reply, but some of those nodes will cancel if they overhear a reply message from any of its covering node. Let $N_2(u)$ denote u's neighbor in graph G_2. Now, for every node $w \in N_2(u)$ it is true that $\xi_{(u,w)} = \emptyset$. In other words, there is no covering node for w. Hence, every node $w \in N_2(u)$ will never cancel its reply message. As a result, all nodes in $N_2(u)$ will be included in the final neighborset of u. Therefore the *Low OverHead Graph* will contain G_2.

(b) The proof follows directly from (a). As G_2 is connected and LOHG contains G_2, it follows that LOHG is connected.

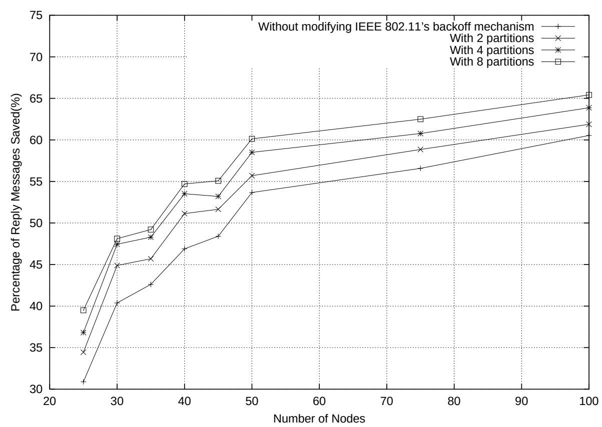
5. EXPERIMENTAL RESULTS

To evaluate the performance of our algorithm, we used a detailed simulation model based on *ns*-2 [1] with wireless extensions. The distributed coordination function (DCF) of the IEEE standard 802.11 [4], was used as the MAC layer. The radio model characteristics were similar to Lucent's WaveLAN [22].

Initially, we deployed $25 - 300$ nodes over a flat square area of $670m \times 670m$. Figure 10(a)



(a) With Equal-length Partition



(b) With Equal-Area Partition

Figure 11. Savings versus number of nodes

shows a typical deployment of 75 nodes, each node having a maximum communication range of $250m$. This is the starting graph G for our algorithm. The remaining parts of Figure 10 show the subgraph G_2, LOHG, and the graph found by R&M. Needless to say, all three subgraphs have considerably fewer links and a lower average node degree than the original maximum powered graph. LOHG has only 2.84% more links than G_2, while the R&M graph has 4.26% more links than G_2.

We ran experiments to see the effect of the varying partition size on the performance of BICOMP, specifically the ability of the biased backoff function to assist the algorithm in prioritizing the reply messages. The performance metrics of interest was the *Saving Ratio* defined as follows:

$$\text{Saving Ratio} = \frac{N_{cancel}}{N_{sent} + N_{cancel} + N_{dropped}} \times 100(\%) \ ,$$

where N_{sent} is the total number of reply messages sent for each NDM requests, N_{cancel} is the number of messages that have been canceled because they were found redundant before transmission, and $N_{dropped}$ is the number of packets dropped by the MAC layer.[†]

Figure 11 shows the *Saving Ratio* for different node density. Each point in the figure is the average from 5 experiments using different samples with the same number of nodes. Three different numbers of partitions, 2, 4 and 8, were considered. The standard backoff mechanism is also included for reference.

The savings appear to be considerably higher for denser networks. Finer partitions also tend to exhibit slightly better performance. The total number of canceled replies was between 30 and 68%.

Figure 12 compares the two partitioning schemes. The *Equal-Area* partitioning seems to slightly outperform the other scheme.

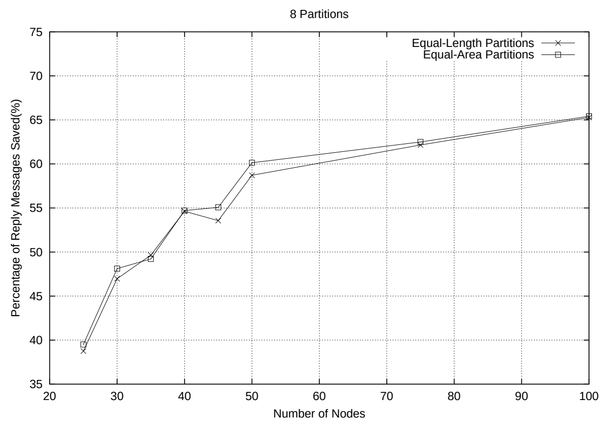


Figure 12. Comparing two partitioning scheme

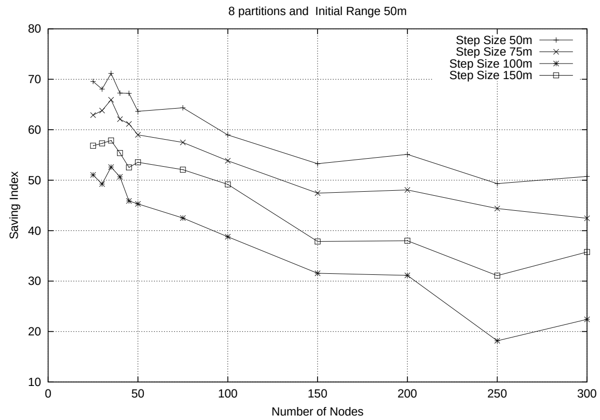
Figure 13 compares the performance of BICOMP with RNSA. The *Saving Index* is defined as follows:

$$\text{Saving Index} = \frac{N_{RNSA} - N}{N_{RNSA}} \times 100(\%) \ ,$$

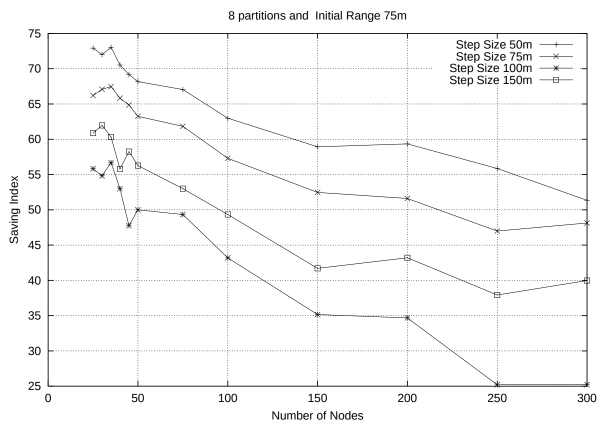
where N is the total number of reply messages needed by our algorithm to construct LOHG, and N_{RNSA} is the corresponding number of messages observed in RNSA. The different graphs correspond to different initial transmission range settings for RNSA (translated from the initial transmission power), and the different curves in each graph correspond to different power increments. The number of partitions used by BICOMP was 8 in all cases.

For sparse networks, the *Saving Index* is very high and drops with the increasing density of nodes. In particular, $35 - 75\%$ of reply messages were eliminated with our approach for less then 50 nodes, and $25 - 55\%$ savings were observed for networks between 50 and 100 nodes. In

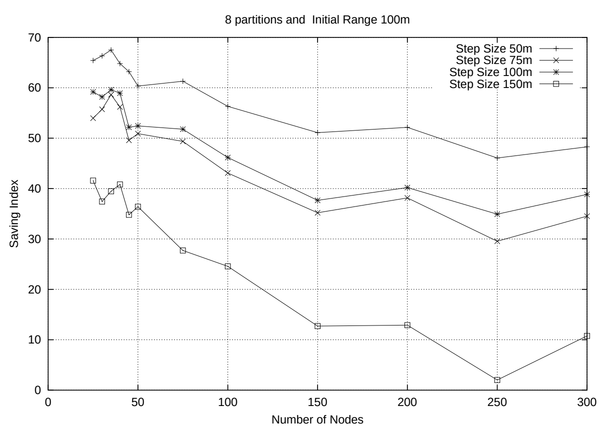
[†]According to IEEE 802.11, after a certain number of unsuccessful transmission attempts the MAC layer drops the packet.



(a)



(b)



(c)

Figure 13. BICOMP versus RNSA

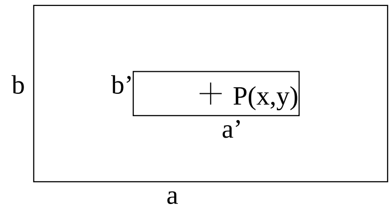


Figure 14. Zipf Distribution

some cases the Saving Index became close to zero. This means that under favorable conditions (if the network is dense enough), RNSA can perform well, but in a sparse or moderately dense network, our approach always brings about a significant improvement.

Finally, we experimented with a Zipf-like distribution of nodes, to see what happens when the network layout departs from uniformity. This study is important because the benefits of the bias in resolving contention during the discovery phase of BICOMP have been argued assuming a more or less balanced structure of a node's neighborhood. Following the standard Zipf bias, we assumed that 80% of nodes are distributed over 20% of total deployment area and the remaining 20% nodes are distributed over the remaining 80% of the deployment area. For a more formal description, consider Figure 14. The total deployment area is the larger rectangular region with the dimensions a and b. Let $P(x, y)$ be the focal point of the distribution. The smaller rectangle centered at P has dimensions a' and b' such that

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

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

Figure 15 shows a typical topology reduction scenario involving 75 nodes under this biased distribution of nodes. The maximum communication range of each node was $250m$.

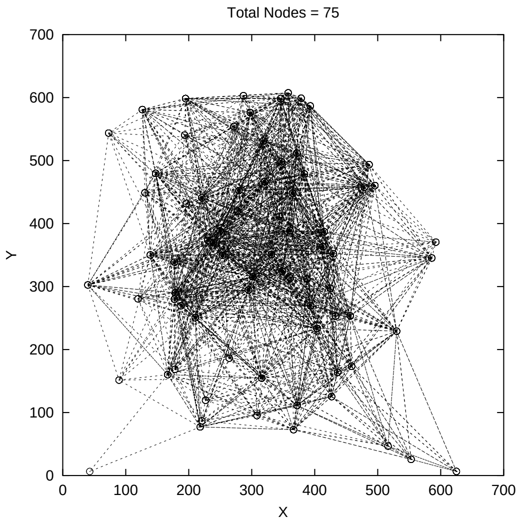
In Figure 16, we show the observed Saving Ratio with different node density under Zipf distribution. Again, three different numbers of partitions, 2, 4 and 8, were considered.

Note that this time the savings of BICOMP for sparse networks are even higher. This seems to dispel our worries that biased distributions may be detrimental to the advantages of biased contention.

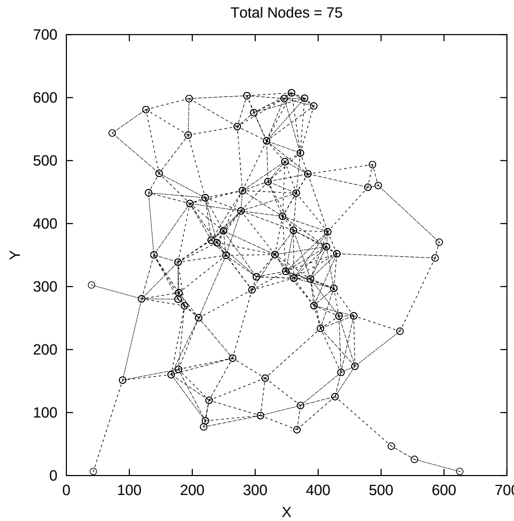
6. CONCLUSIONS

We have presented a MAC-assisted algorithm for constructing minimum energy path preserving graphs in ad-hoc wireless networks. Our studies have demonstrated the superiority of the new algorithm over the previous solution for networks with moderate and low density of nodes.

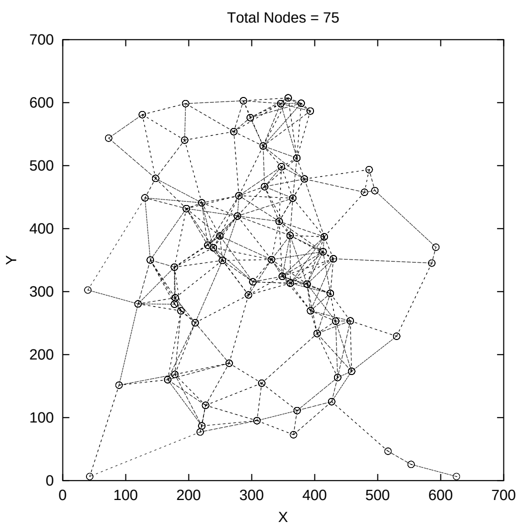
The notion of minimum-energy path preserving graphs is important from the viewpoint of network performance, even if we completely ignore the power saving gains. It gives a natural and rational way of reducing the degree of the network graph, which allows the nodes to reduce



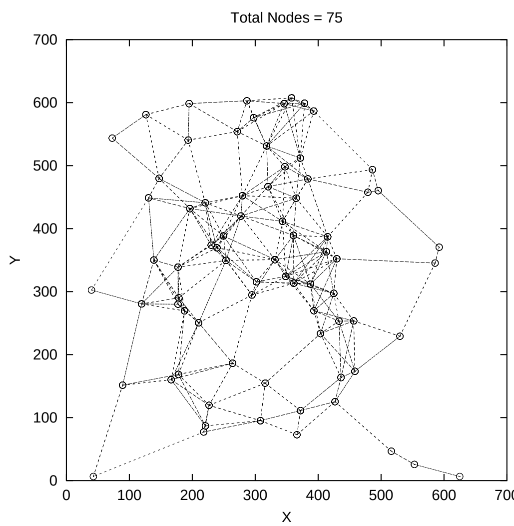
(a) Original graph



(b) G_2 generated by RNSA

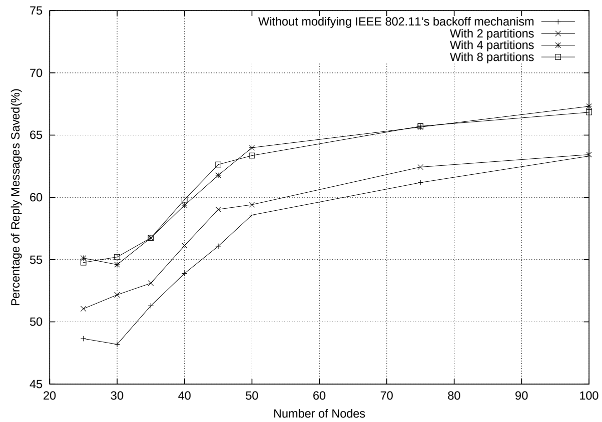


(c) LOHG

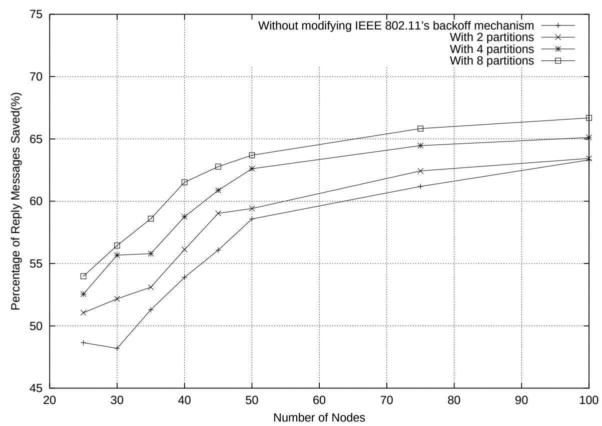


(d) Generated by R&M

Figure 15. Reduced graphs under Zipf distribution of nodes



(a) With Equal-length Partition



(b) With Equal-Area Partition

Figure 16. Savings of BICOMP under Zipf distribution

the number of neighbors they have to talk to and thus reduce the overall contention to the scarce radio channel.

Our exercise demonstrates once again that strict protocol layering is a curse of wireless networks. The issue of power control calls for the collaboration of all layers and keeping some layers closed may significant impair the flexibility of the whole protocol stack. One would like to see more parameterization in the medium access layer that would make it possible to modify the contention resolution algorithms from the routing (network) layer and, possibly, from the application.

ACKNOWLEDGEMENT

The authors would like to thank Joseph Halpern and Li Li (the authors of [8]), for the valuable

clarification of their algorithm.

REFERENCES

1. The Network Simulator: NS-2: notes and documentation. http://www.isi.edu/nsnam/ns/.
2. M. Burkhart, P. von Rickenbach, R. Wattenhofer, and A. Zollinger. Does topology control reduce interference? In *Proceedings of ACM MobiCom*, pages 9–19, 2004.
3. Dr-Jiunn Deng and Ruay-Shiung Chang. A priority scheme for IEEE 802.11 DCF acess method. *IEICE Trans. Commun.*, E82-B(1), January 1999.
4. IEEE Standards Department. Wireless LAN medium access control (MAC) and physical layer (PHY) specifications, 1997. IEEE standard 802.11-1997.
5. Z. Huang, C. Shen, C. Srisathapornphat, and C. Jaikaeo. Topology control for ad hoc networks with directional antennas. In *Proc. IEEE Int. Conference on Computer Communications and Networks*, pages 16–21, 2002.
6. V. Kawadia and P. R. Kumar. Power control and clustering in ad hoc networks. In *Proceedings of INFOCOM 2003*, San Francisco, USA, April 2003.
7. V. Kawadia and P. R. Kumar. Principles and protocols for power control in ad hoc networks. *IEEE Journal on Selected Areas in Communications*, 23(5):76–88, January 2005.
8. L. Li and J. Halpern. Minimum energy mobile wireless networks revisited. In *Proceedings of IEEE Conference of Communications (ICC '01)*, 2001.
9. L. Li, J. Y. Hlapern, P. Bahl, Y. Wang, and R. Watenhofer. Analysis of a con-based topology control algorithm for wireless multi-hop networks. In *In ACM Symposium on Principle of Distributed Computing (PODC)*, 2001.
10. N. Li, J. C. Hou, and L. Sha. Design and analysis of an MST-based topology control algorithm. In *Proceedings of INFOCOM*, 2003.
11. K. Moaveninejad and X. Y. Li. Low-interference topology control for wireless ad hoc networks, 2005. Accepted for publication.
12. J. P. Monks, V. Bhargavan, and W. M. Hwu. A power controlled MAC protocol for wireless packet networks. In *Proceedings of INFOCOM*, pages 219–228, 2001.
13. S. Narayanaswamy, V. Kawadia, R. S. Sreenivas, and P. R. Kumar. Power control in ad-hoc networks: Theory, architecture, algorithm and implementation of the COMPOW protocol. In *Proceedings of the European Wireless Conference - Next Generation Wireless Networks: Technologies, Protocols, Services and Applications*, pages 156–162, Florence, Italy, February 2002.
14. A. Rahman, W. Olesinski, and P. Gburzynski. Controlled flooding in wireless ad-hoc networks. In *Proceedings of IWWAN'04*, Oulu, Finland, jun 2004.
15. R. Ramanathan and R. Rosales-Hain. Topology control of multihop wireless networks using transmit power adjustment. In *In Proc. of IEEE INFOCOM 2000*, pages 404–413, Tel Aviv, Israel, March 2000.
16. T.S. Rappaport. Wireless communications: principles and practice, 1996. Prentice Hall.
17. V. Rodoplu and T. Meng. Minimum energy mobile wireless networks. *IEEE Journal of Selected Areas in Communications*, 17(8):1333–1344, 1999.
18. S. Singh and C.S. Raghavendra. Power efficient MAC protocol for multihop radio networks. In *Proceedings of PIMRC*, pages 153–157, 1998.
19. S. Singh, M. Woo, and C.S. Raghavendra. Power-aware routing in mobile ad hoc networks. In *Proceedings of MobiCom*, pages 181–190, 1998.
20. Yuh-Shyan C hen Sze-Yao Ni, Yu-Chee Tseng and Jang-Ping Sheu. The broadcast storm problem in a mobile ad hoc network. In *Mobicom*, 1999.
21. J. Tang, G. Xue, and W. Zhang. Interference-aware topology control and qos routing in multi-channel wireless mesh networks. In *Proceedings of ACM MobiHoc*, pages 68–77, Urbana-Champaign, Illinois, USA, May 2005.
22. B. Tuch. Development of WaveLAN, an ISM band wireless LAN. *AT&T Technical Journal*, 72(4):27–33, July/Aug 1973.
23. R. Wattenhofer, L. Li, P. Bahl, and Y.-M. Wang. Distributed topology control for wireless multihop ad-hoc networks. In *INFOCOM*, pages 1388–1397, 2001.
24. A. Yao. On constructing minimum spanning trees in k-dimensional spaces and related problems. *SIAM Journal on Computing*, pages 721–736, 1982.

AUTHORS' BIOGRAPHIES

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 is currently a PhD candidate in the Department of Computing Science, University of Alberta. His research interests include ad-hoc and sensor networks, peer-to-peer computing, swarm intelligence, back-end compiler optimization and neural networks.

Pawel Gburzynski received his MSc and PhD in Computer Science from the University of Warsaw, Poland in 1976 and 1982, respectively. Before coming to Canada in 1984, he had been a research associate, systems programmer, and consultant in the Department of Mathematics, Informatics, and Mechanics at the University of Warsaw. Since 1985, he has been with the Department of Computing Science, University of Alberta, where he is a Professor. Dr. Gburzynski's research interests are in communication networks, operating systems, simulation, and performance evaluation.