Energy Efficient Broadcasting Using Sparse Topology in Wireless Ad Hoc Networks

Kousha Moaveninejad* Xiang-Yang Li*

Keywords—**Wireless ad hoc networks, broadcasting, power consumption, spare structures, topology control.**

I. Introduction

Wireless Ad Hoc Networks: Due to its potential applications in various situations such as battlefield, emergency relief, environment monitoring, and so on, wireless ad hoc networks [1], [2], [3], [4] have recently emerged as a premier research topic. Wireless networks consist of a set of wireless nodes which are spread over a geographical area. These nodes are able to perform processing as well as capable of communicating with each other by means of a wireless ad hoc network. With coordination among these wireless nodes, the network together will achieve a larger task both in urban environments and in inhospitable terrain. For example, the sheer numbers of wireless sensors and the expected dynamics in these environments present unique challenges in the design of wireless sensor networks. Many excellent researches have been conducted to study problems in this new field [1], [2], [5], [3], [6], [4].

In this paper, we consider a wireless ad hoc network consisting of a set V of n wireless nodes distributed in a two-dimensional plane. Each wireless node has an omnidirectional antenna. This is attractive because a single transmission of a node can be received by many nodes within its vicinity which, we assume, is a disk centered at the node. This property is also known as "*wireless multicast advantage*". We call the radius of this disk the *transmission range* of this wireless node. In other words, node v can receive the signal from node u if node v is within the transmission range of the sender u. Otherwise, two nodes communicate through multi-hop wireless links by using intermediate nodes to relay the message. Consequently, each node in the wireless network also acts as a router, forwarding data packets for other nodes. By a proper scaling, we assume that all nodes have the maximum transmission range equal to one unit. These wireless nodes define a *unit disk graph* $\mathrm{UDG}(V)$ in which there is an edge between two nodes if and only if their Euclidean distance is less than or equal to one.

In addition, we assume that each node has a low-power Global Position System (GPS) receiver, which provides the position information of the node itself. If GPS is not available, the distance between neighboring nodes can be estimated on the basis of incoming signal strengths. Relative co-ordinates of neighboring nodes can be obtained by exchanging such information between neighbors [7]. With the position information, we can apply computational geometry techniques to solve some challenging questions in wireless networks.

*Department of Computer Science, Illinois Institute of Technology, 10 W. 31st Street, Chicago, IL 60616, USA. Email moavkoo@iit.edu, xli@cs.iit.edu.

Power-Attenuation Model: Energy conservation is a critical issue in wireless network for the node and network life, since the nodes are powered by batteries only. Each wireless node typically has a portable set with transmission and reception processing capabilities. To transmit a signal from a node to the other node, the power consumed by these two nodes consists of the following three parts. First, the source node needs to consume some power to prepare the signal. Second, in the most common power-attenuation model, the power needed to support a link uv is $\|uv\|^\beta$, where $\|uv\|$ is the Euclidean distance between u and v, β is a real constant between 2 and 5 dependent on the transmission environment. This power consumption is typically called *path loss*. Finally, when a node receives the signal, it needs to consume some power to receive, store and then process that signal. For simplicity, this overhead cost can be integrated into one cost, which is almost the same for all nodes. Thus, we will use c to denote such constant overhead. In most results surveyed here, it is assumed that $c = 0$, i.e., the path loss is the major part of power consumption to transmit signals. The power cost $p(e)$ of a link $e = uv$ is then defined as the power consumed for transmitting signal from u to node v, i.e. $p(uv) = \|uv\|^\beta$.

Broadcast, Multicast and Unicast: Broadcasting is a communication paradigm that allows sending data packets from a source node to all nodes in the network, Multicasting is a communication paradigm that allows sending data packets from a source node to multiple receivers and Unicasting is a communication paradigm that allows sending data packets from a source node to a single destination node. In one-to-all model, (using omni-directional antennas) transmission by each node can reach *all* nodes that are within radius distance from it, while in the one-to-one model, each transmission is directed toward only one neighbor (using, for instance, directional antennas or separate frequencies for each node). ***The broadcasting in literature has been studied mainly for one-to-all model and we will use that model in this chapter. Broadcasting is also frequently referred to as flooding.***

Broadcasting and multicasting in wireless ad hoc networks are critical mechanisms in various applications such as information diffusion, wireless networks, and also for maintaining consistent global network information. Broadcasting is often necessary in MANET routing protocols. For example, many unicast routing protocols such as Dynamic Source Routing (DSR), Ad Hoc On Demand Distance Vector (AODV), Zone Routing Protocol (ZRP), and Location Aided Routing (LAR) use broadcasting or a derivation of it to establish routes.

Currently, most of the protocols rely on a simplistic form of broadcasting called *Flooding*, in which each node (or all nodes in a localized area) retransmits each received unique packet exactly one time. The main problems with Flooding are that it typically causes unproductive and often harmful bandwidth congestion, as well as inefficient use of node resources. Broadcasting is also more efficient than sending multiple copies of the same packet through unicast. It is highly important to use power-efficient broadcast algorithms for such networks since***, as mentioned before,*** wireless devises are often powered by batteries only.

Recently, a number of research groups have proposed more efficient broadcasting techniques [8], [9], [10], [11], [12], [13], [14] with various goals such as minimizing the number of retransmissions, minimizing the total power used by all transmitting nodes, minimizing the overall delay of the broadcasting, and so on. Williams and Camp [13] classified the broadcast protocols into four categories: simple (blind) flooding, probability based, area based, and neighbor knowledge methods. Wu and Lou [15] classified broadcasting protocols based on neighbor knowledge information ***into four categories***: global, quasi-global, quasi-local, and local. The global broadcast protocol, ***which could be*** centralized or distributed, is based on global state information. In quasi-global broadcasting, a broadcast protocol is based on partial global state information. For example, the approximation algorithm in [16] is based on building a global spanning tree (a form of partial global state information) that is constructed in a sequence of sequential propagations. In quasi-local broadcasting, a distributed broadcast protocol is based on mainly local state information and occasionally partial global state information. Cluster networks are such examples: while clusters can be constructed locally for most of the time, the chain reaction does occur occasionally. In local broadcasting, a distributed broadcast protocol is solely based on local state information. All protocols that select forward nodes locally (based on 1-hop or 2-hop neighbor set) belong to this category. It has been recognized that scalability in wireless networks cannot be achieved by relying on solutions where each node requires global knowledge about the network. To achieve scalability, the concept of localized algorithms was proposed, as distributed algorithms where simple local node behavior, based on local knowledge, achieves a desired global objective.

MAC Specification: Collision avoidance is inherently difficult in MANETs; one often cited difficulty is overcoming the hidden node problem, where a node cannot decide whether some of its neighbors are busy receiving transmissions from an uncommon neighbor. The 802.11 MAC follows a Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA) scheme. For unicasting, it utilizes a Request To Send (RTS) / Clear To Send (CTS) / Data / Acknowledgment (ACK) procedure to account for the hidden node problem. However, the RTS/CTS/Data/ACK procedure is too cumbersome to implement for broadcast packets as it would be difficult to coordinate and bandwidth expensive: a relay node has to perform RTS/CTS individually with all its neighbors that should receive the packets. Thus, the only requirement made for broadcasting nodes is that they assess a clear channel before broadcasting. Unfortunately, clear channel assessment does not prevent collisions from hidden nodes. Additionally, no resource is provided for collision when two neighbors assess a clear channel and transmit simultaneously. Ramifications of this environment are subtle but significant. Unless specific means are implemented at the network layer, a node has no way of knowing whether a packet was successfully reached by its neighbors. In congested networks, a significant amount of collisions occur leading to many dropped packets. The most effective broadcasting protocols try to limit the probability of collisions by limiting the number of rebroadcasts in the network. Thus, it is often imperative the underlying structure for broadcasting is degree bounded and the links are at similar lengths. By using a power adjustment at each node, the collision of packets and contention for channel will be alleviated. Notice that, if the underlying structure for broadcasting is degree bounded, we can either use RTS/CTS scheme to avoid hidden node problem, or we can rebroadcast the dropped packets (such rebroadcast will be less since the number of intended receiving neighbors is bounded by a small constant).

Performance Measurement: The performance of broadcast protocols can be measured by variety of metrics. A commonly used metric is the number of message retransmissions with respect to the number of nodes. In case of broadcasting with *adjustable transmission power*, which will be explained later in section II, the total power is used as performance metrics, while In case of broadcasting with *non-adjustable transmission power*, which also will be explained later in section II, the number of forwarding nodes, also known as dominators, is used as performance metrics

Organization The rest of the chapter is organized as follows. In Section II, we review the priori arts of energy efficient broadcasting based on structures constructed in centralized manner or localized manner. In Section III we study broadcasting based on sparse topology constructed locally and to validate our theoretical results we conducted extensive simulations in Section III. We conclude the paper in Section VI.

II. Preliminaries

Network Model: We assume that all wireless nodes are given as a set V of n points in a two dimensional space and each wireless node has some computational power and an omni-directional antenna. This is attractive because a single transmission by a wireless node can be received by all wireless nodes within its vicinity. It is also assumed that the nodes are almost static in a reasonable period of time, all wireless nodes have distinctive identities and each wireless node u has a maximum transmission range R_u. A directed communication graph $\overrightarrow{G} = (V, \overrightarrow{E})$ over a set V of wireless nodes has an edge $\overrightarrow{uv}$ from node u to node v

if and only if node u can send message directly to node v (i.e., $\|uv\| \leq R_u$). If the maximum transmission range of all wireless nodes are the same, then all communications edge will be mutual, i.e., $\overrightarrow{uv}$ exists iff $\overrightarrow{vu}$ exists, and we can ignore the direction of edges and by proper scaling we can set the maximum transmission of all wireless nodes to one unit and model the graph as *UDG(Unit Disk Graph)*.Unit Disk Graph is an undirected graph where there is an edge between two nodes if and only if the Euclidean distance between them is less than one unit (See Figure 1(a) for an illustration). We always assume the network is connected, otherwise sending message from the source node to all the nodes in the network would be impossible.

Minimum Connected Dominating Set:A subset S of V is a *dominating set* if each node u in V is either in S or is adjacent to some node v in S. Nodes from S are called *dominators*, while nodes not is S are called *dominatees*. A subset C of V is a *connected dominating set* (CDS) if C is a dominating set and C induces a connected subgraph. Consequently, the nodes in C can communicate with each other without using nodes in $V - C$. A dominating set with minimum cardinality is called *minimum dominating set*, denoted by MDS. A connected dominating set with minimum cardinality is denoted by *minimum connected dominating set* (MCDS). A broadcasting based on connected dominating set only uses the nodes in CDS to relay the message.

In Figure 1 dominators are shown by squares. Figure 1(a) shows the original UDG graph and as can be seen in Figure 1(b), all nodes in the graph are either in the backbone or at least have a neighbor in the backbone. Figure 1(c) shows the backbone of the same graph.

Broadcast Tree: Any broadcast routing can be viewed as an arborescence (a directed tree) T, rooted at the source node of the broadcasting, that spans all nodes. Let $f_T(\mathbf{p})$ denote the transmission power of the node $\mathbf{p}$ required by T. For any leaf node $\mathbf{p}$ of T, $f_T(\mathbf{p}) = 0$. For any internal node $\mathbf{p}$ of T, $f_T(\mathbf{p})$ depends on the power model that is used for broadcasting.

In Figure 2(a) a message from source node s is broadcasted to all nodes in the network. Dashed circles represent the transmission area of node s,1 and 2. Figure 2(b) shows an isomorphic arborescence to what is shown in Figure 2(a). As can be seen leaf nodes, (i.e., nodes $2, 3, 4, 6, 7$, and 8 in this examples) do not contribute in broadcast and consume no energy.

Power Model:In the literature, there are two common energy models that could be used for broadcasting:

- *Non-adjustable power*: In this model, each node uses its maximum transmission range to send message or data to the nodes in its vicinity. In other words, the power consumed at each node is not adjustable and is a constant for all relay nodes. Since we assumed the maximum transmission range of all nodes is 1 unit, for any internal node p in the broadcast tree T we have:

$$f_T(\mathbf{p}) = 1,$$

So minimizing the total power used by a reliable broadcast tree is equivalent to the minimum connected dominating set problem (MCDS) (i.e., minimize the number of nodes that relay the message), since all relaying nodes of a reliable broadcast form a connected dominating set (CDS).

- *Adjustable power*: In this model the power consumed at each node is adjustable. we assume that the power consumed by a relay node u is $\|uv\|^\beta$, where the real number $\beta \in [2, 5]$ depending on transmission environment and v is the farthest neighbor of u in the broadcast tree. For any internal node $\mathbf{p}$ of T,

$$f_T(\mathbf{p}) = \max_{\mathbf{pq} \in \mathbf{T}} \|\mathbf{pq}\|^\beta ,$$

in other words, the β-th power of the longest distance between $\mathbf{p}$ and its children in T. The total energy required by T is $\sum_{\mathbf{p} \in \mathbf{P}} f_T(\mathbf{p})$.

In the rest of this chapter, for these two energy models respectively, we will review several methods that can build some broadcast trees whose energy consumption are within a constant factor of the optimum if the original communication graph is modelled by unit disk graph.

Approximation ratio: Approximation ratio of a heuristic of a minimization problem is the maximum ratio of value given by the heuristic to value of the answer of the problem. So for minimization problem A, and heuristic H, Approximation ratio α is defined as:

$$\alpha =_{SUP} \frac{A(H)}{A(optimal)}$$

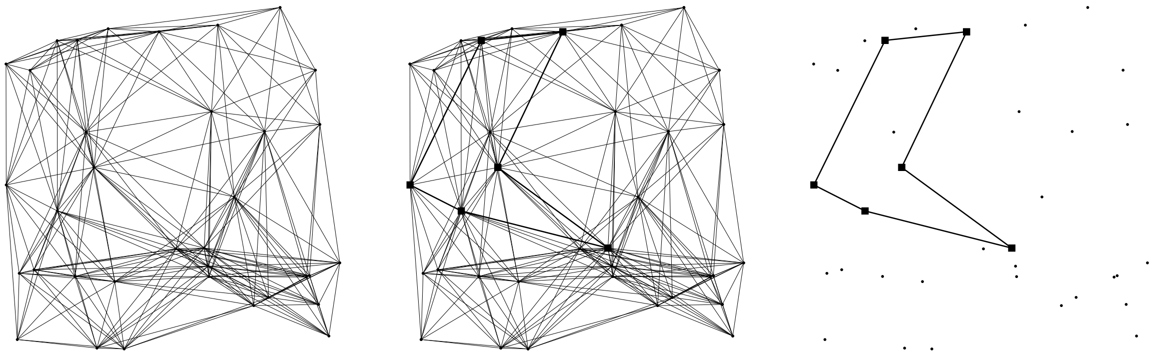
and Approximation ratio of a heuristic of a maximization problem is the maximum ratio of the value of the optimal answer of the problem to the value given by the heuristic.So for maximization problem A, and heuristic H, Approximation ratio α is defined as:

$$\alpha =_{SUP} \frac{A(optimal)}{A(H)}$$

A. *Centralized Methods*

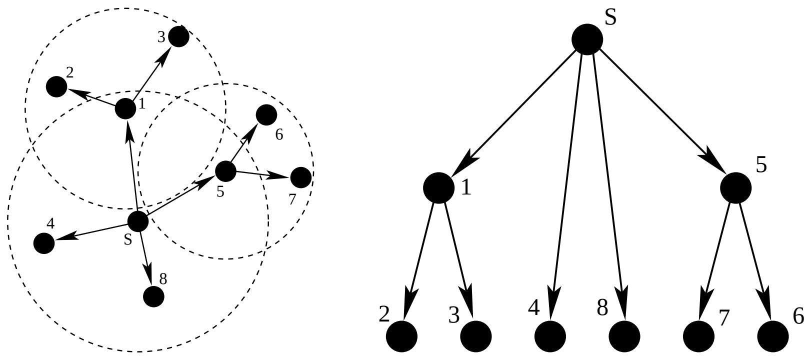
A.1 Based on adjustable power model:MST and Variations

Some centralized methods are based on greedy heuristics. Three greedy heuristics were proposed in [17] for the minimum-energy broadcast routing problem: MST (minimum spanning tree), SPT (shortest-path tree), and BIP (broadcasting incremental power). MST is the tree that spans all the nodes and has the minimum total edge length. The MST heuristic first applies the Prim's algorithm to obtain a MST, and then orients it as an arborescence rooted at the source node. SPT is the tree that spans all the nodes such that the shortest path between every pair of nodes is included. In other words, for any pairs of nodes u and v, the shortest Euclidean path that connects node u to node v belongs to SPT. The SPT heuristic applies the Dijkstra's algorithm to obtain a SPT rooted at the source node. The BIP heuristic is the node version of Dijkstra's algorithm for SPT. It maintains, throughout its execution, a single arborescence rooted at the source node. The arborescence starts from the source node, and new nodes are



(a) (b) (c)

Fig. 1. UDG and CDS example.



(a) (b)

Fig. 2. Broadcast tree example.

added to the arborescence one at a time on the minimum incremental cost basis until all nodes are included in the arborescence. The incremental cost of adding a new node to the arborescence is the minimum additional power increased by some node in the current arborescence to reach this new node.

The minimum-energy broadcast routing problem is different from the conventional link-based minimum spanning tree (MST) problem. Minimum-energy broadcast routing problem finds the tree such that the total power consumed by the node in the broadcast tree in minimized while the conventional link-based minimum spanning tree (MST) problem finds the tree such that the total Euclidean edge length is minimized.

Indeed, while the MST can be solved in polynomial time by algorithms such as Prim's algorithm and Kruskal's algorithm [18], the minimum-energy broadcast routing problem cannot be solved in polynomial time unless P=NP [19]. Recently, Clementi *et al.* [19] proved that the minimum-energy broadcast routing problem is NP-hard and obtained a parallel but weaker result to those of [20].

Wan *et al.* [20] showed that the approximation ratios of MST and BIP are between 6 and 12 and between $\frac{13}{3}$ and 12 respectively.

here approximation ratio of a heuristic is the maximum ratio of the energy needed to broadcast a message based on the arborescence generated by this heuristic to the least necessary energy by any arborescence for any set of points.

Another two greedy heuristics were proposed in [17] for the minimum-energy broadcast routing problem: BLU and BLiMST. BLU (Broadcast Least-Unicast-cost) algorithm is a straightforward (but far from optimal) approach. BLU builds the broadcast trees that consist of the superposition of the best unicast paths to each individual destination. It is assumed that an underlying unicast algorithm (such as the Bellman-Ford or Dijkstra algorithm) provides "minimum-distance" paths from the source node to every other node. Since BLU is based on the use of a scalable unicast algorithm, it also is scalable.

Also note that, although algorithms based on minimum distance paths are normally used for packet-switched applications, this approach is being used here for session oriented traffic, since a cost (involving power and possibly congestion) can be defined for each link in the network. By contrast, in circuit-switched wired applications it is difficult to define a link cost because energy is not of concern and because delay is not an appropriate metric (as it would be in packet-switched applications) since resources are reserved in circuit-switched applications. Instead, blocking probability is the only overall objective, and there is no known way of

mapping that objective to individual link metrics.

Summarizing the above, we have:

BLU: A minimum-cost path from the source node to every other node is established. The broadcast tree consists of the superposition of these unicast paths.

The failure of BLU to exploit the wireless multicast advantage results in higher overall power expenditure.

BLiMST (Broadcast Link-based MST) based on the use of the standard MST formulation (as in wired networks) in which a link cost is associated with each pair of nodes (i.e., the power to sustain the link). Thus, the "wireless multicast advantage" is ignored in the construction of the MST. Since the MST problem is of polynomial complexity, it is scalable. Once the MST is constructed in this manner, the evaluation of its cost (i.e., the total power needed to sustain the broadcast tree) does take into consideration the wireless multicast advantage.

Summarizing the above, we have:

BLiMST: A minimum-cost (minimum-power) spanning tree is formed using standard (link-based) MST techniques.

Similar to the case of BLU, the failure of BLiMST to exploit the wireless multicast advantage results in higher overall power expenditure. The complexity of BLU, when implemented by means of the Dijkstra algorithm, is $O(N^2)$, where N is the number of nodes in the network. The complexity of BLiMST, when implemented by means of Prim's algorithm, is $O(N^3)$ when a straightforward implementation is used. However, a more sophisticated implementation using a Fibonacci heap yields complexity $O(M + NlogN) = O(N^2)$, where $M = \frac{N(N-1)}{2}$ is the number of links (in a fully connected network). Since BIP is based on Prim's algorithm, it also has complexity $O(N^3)$.

The Sweep: Removing Unnecessary Transmissions:The performance of the algorithms presented here can be improved by eliminating unnecessary transmissions by means of what we call the "sweep" operation. The sweep procedure is summarized as follows. We examine the nodes in ascending ID order, Leaf nodes are ignored because they do not transmit. Each nodes whose neighbors are all covered by nodes with lower ID's will become a leaf node. It is easy to show that sweep operation doesn't disconnect the broadcast tree.

Typically, a single application of the sweep operation provides significant improvement; small further improvement can often be obtained by repeating the sweep once more, but little improvement has been found by additional applications of this procedure. However, in most cases the tree produced by BIP has lower power, both before and after the sweep.

A.2 Based on non-adjustable power model:Clustering

ADD WU and LI's method

We now study the non-adjustable power model case. When using the non-adjustable model, every internal node in the broadcast tree (also called relay node) uses its maximum power, so the power consumed at all internal nodes are equal. In other words for every pair of internal nodes p and q in the broadcast tree T we have:

$$f_T(\mathbf{p}) = f_T(\mathbf{q}),$$

Since the energy consumed at all relay nodes are equal, our goal is to minimize the number of relay nodes in the broadcast tree, so the set of nodes that rebroadcast the message in a reliable broadcasting scheme define a connected dominating set.

In other words, if nodes cannot adjust their transmission powers accordingly, then we need to find the minimum connected dominating set to save the total power consumption of the broadcasting protocol. Unfortunately, the problem of finding connected dominating set of minimal size is NP-complete even for unit disk graphs.

We first review several methods in the literature to build a connected dominating set.

Notice that, Berman *et al.* [21] gave an $\frac{4}{3}$ approximation method to connect a dominating set and Robins *et al.* [22] gave an $\frac{4}{3}$ approximation method to connect an independent set and a PTAS for minimum dominating set was reported in [23]. Thus, we can easily have an $\frac{8}{3}$ approximation algorithm for MCDS, which was reported in [24]. Recently, Cheng *et al.* [25] designed a PTAS for MCDS in UDG. However, it is impossible to run their method efficiently in a distributed manner. Several distributed clustering (or dominating set) algorithms have been proposed in the literature [26], [27], [28], [29], [30], [31]. All algorithms assume that the nodes have distinctive identities (denoted by ID hereafter).

B. Localized Methods

The centralized algorithms do not consider computational and message overheads incurred in collecting global information. Several of them also assume that the network topology does not change between two runs of information exchange. These assumptions may not hold in practice, since the network topology may change from time to time, and the computational and energy overheads incurred in collecting global information may not be negligible. This is especially true for large-scale wireless networks where the topology is changing dynamically due to the changes of position, energy availability, environmental interference, and failures, which implies that centralized algorithms that require global topological information may not be practical.

Some distributed heuristics are proposed, such as [32], [33], [34]. Most of them are based on distributed MST method. A possible drawback of these distributed methods is that they may not perform well under frequent topological changes as they rely on information that is multiple hops away to construct the MST. Refer to [35] for more detail. Localized minimum energy broadcast algorithms are based on the use of a locally defined geometric structures, such as RNG (relative neighborhood graph), proposed by Toussaint [36]. RNG consists of all edges uv such that uv is not the longest edge in any triangle uvw. That is, uv belongs to RNG if there is no node w such that $uw < uv$ and $vw < uv$. Cartigny *et al.* [37] proposed a localized algorithm, called RBOP [37] that is built upon the notion of relative neighborhood graph (RNG) using the rules of neighbor elimination [38]. Simulation results show that the energy consumption could be as high as 100% compared to BIP.

Li and Hou [35], and Cartigny *et al.* [39] proposed another localized algorithm, which applies LMST (localized minimum spanning tree) instead of RNG as the broadcast topology. In LMST, proposed in [40], each node calculates local minimum spanning tree of itself and its 1-hop neighbors. A node uv is in LMST if and only if u and v select each other in their respective trees. The simulations [35], [39] show that the performance of LMST based schemes is significantly better than the performance of RBOP, and with about 50% more energy consumption than BIP in static scenarios.

However, as shown in [41], the total energy used based on RNG and LMST could still be as large as $O(n^2)$ times of the total energy used by MST. Given a graph G, let $\omega_b(G) = \sum_{e \in G} \|e\|^b$. Then $\omega_b(RNG) = \Theta(n^b) \cdot \omega_b(MST)$ and $\omega_b(LMST) = \Theta(n^b) \cdot \omega_b(MST)$. In [41], [42], we described three low weight planar structures that can be constructed by localized methods with total communication costs $O(n)$. The energy consumption of broadcast based on those structures are within $O(n^{\beta-1})$ of the optimum, which improves the previously known "lightest" structure RNG by $O(n)$ factor.

C. Flooding Based Methods

The simplest broadcasting mechanism is to let every node retransmit the message to all its one-hop neighbors when receiving the first copy of the message, which is called *flooding* in the literature. Despite its simplicity, flooding is very inefficient and can result in high redundancy, contention, and collision. One approach to reduce the redundancy is to let a node only forward the message to a subset of one-hop neighbors who together can cover the two-hop neighbors. In other words, when a node retransmits a message to its neighbors, it explicitly asks a subset of its neighbors to relay the message.

In [**?**], Lim and Kim proposed a broadcasting scheme that chooses some or all of its one-hop neighbors as rebroadcasting node. When a node receives a broadcast packet, it uses a Greedy Set Cover algorithm to determine which subset of neighbors should rebroadcast the packet, given knowledge of which neighbors have already been covered by the sender's broadcast. The Greedy Set Cover algorithm recursively chooses 1-hop neighbors which cover the most 2-hop neighbors and recalculates the cover set until all 2-hop neighbors are covered.

Călinescu *et al.* [43] gave two practical heuristics for this problem (they called selecting forwarding neighbors). The first algorithm runs in time $O(n \log n)$ and returns a subset with size at most 6 times of the minimum. The second algorithm has an improved approximation ratio 3, but with running time $O(n^2)$. Here n is the number of total two-hop neighbors of a node. When all two-hop neighbors are in the same quadrant with respect to the source node, they gave an exact solution in time $O(n^2)$ and a solution with approximation factor 2 in time $O(n \log n)$. Their algorithms partition the region surrounding the source node into four quadrants, solve each quadrants using an algorithm with approximation factor α, and then combine these solutions. They proved that the combined solution is at most 3α times of the optimum solution.

Their approach assumes that every node u can collect its 2-hop neighbors $N_2(u)$ efficiently. Notice that, the 1-hop neighbors of every node u can be collected efficiently by asking each node to broadcast its information to its 1-hop neighbors. Thus all nodes get their 1-hop neighbors information by using total $O(n)$ messages. However, until recently, it was not known how to collect the 2-hop neighbors information with $O(n)$ communications. The simplest broadcasting of 1-hop neighbors $N_1(u)$ to all neighbors u does let all nodes in $N_1(u)$ to collect their corresponding 2-hop neighbors. However, the total communication cost of this approach is $O(m)$, where m is the total number of links in UDG. Recently, Călinescu [44] proposed an efficient approach to collect $N_2(u)$ using the connected dominating set [45] as forwarding nodes. Assume that the node position is known. He proved that the approach takes total communications $O(n)$, which is optimum within a constant factor.

The Probabilistic scheme from [12] is similar to Flooding, except that nodes only rebroadcast with a predetermined probability. When the probability is 100%, this scheme is identical to Flooding.

Cartigny and Simplot [**?**] applied probability which is a function of the distance to the transmitting neighbor. Tseng *et al.* [12] shows an inverse relationship between the number of times a packet is received at a node and the probability of that node being able to reach additional area on a rebroadcast. This result is the basis of their Counter-Based scheme. Upon reception of a previously unseen packet, the node initiates a counter with a value of one and sets a RAD (which is randomly chosen between 0 and $Tmax$ seconds). During the RAD, the counter is incremented by one for each redundant packet received. If the counter is less than a threshold value when the RAD expires, the packet is rebroadcasted. Otherwise, it is simply dropped. From [12], threshold values above six relate to little additional coverage area being reached.

In either probabilistic schemes or the counter-based schemes a node decides whether to rebroadcast a received packet purely based on its own information. Tseng *et al.* [12] proposed several other criteria based on the additional coverage area to decide whether the node will rebroadcast the packet.

These coverage-area based methods are similar to the methods of selecting forwarding neighbors, which tries to select a set of one-hop neighbors sufficient to cover all its two-hop neighbors. While area based methods only consider the coverage area of a transmission; they don't consider whether nodes exist within that area. Two coverage-area based methods are proposed in [12]: *Distance-Based Scheme* and *Location Based Scheme.*

In Distance-Based Scheme, a node compares the distance between itself and each neighbor node that has previously rebroadcast a given packet. Upon reception of a previously unseen packet, a RAD is initiated and redundant packets are cached. When the RAD expires, all source node locations are examined to see if any node is closer than a threshold distance value. If true, the node doesn't rebroadcast.

The Location-Based scheme uses a more precise estimation of expected additional coverage area in the decision to rebroadcast.

In this method, each node must have the means to determine its own location, e.g., a GPS. Whenever a node originates or rebroadcasts a packet it adds its own location to the header of the packet. When a node initially receives a packet, it notes the location of the sender and calculates the additional coverage area obtainable were it to rebroadcast. If the additional area is less than a threshold value, the node will not rebroadcast, and all future receptions of the same packet will be ignored. Otherwise, the node assigns a RAD before delivery. If the node receives a redundant packet during the RAD, it recalculates the additional coverage area and compares that value to the threshold. The area calculation and threshold comparison occur with all redundant broadcasts received until the packet reaches either its scheduled send time or is dropped.

Instead of covering area, one could simply cover neighboring nodes, assuming their location, or existence of their link to a previous transmitting node, are known. The basic method was independently and almost simultaneously (August 2000) proposed in two articles [**?**], [**?**]. The methods were called Neighbor Elimination by Stojmenovic and Seddigh [**?**], while a similar method, called Scalable Broadcast Algorithm, was proposed by Peng and Lu [**?**]. Two-hop neighbors information is used to determine whether a node will rebroadcast the packet. Suppose that a node u receives a broadcast data packet from its neighbor node v. Node u knows all the neighbors of node v, and thus all nodes that are common neighbors of them (already received the data from v). If node u has additional neighbors not reached by node v's broadcast, node u schedules the packet for delivery with a RAD. However, if node u receives a redundant broadcast packet from some other neighbors within RAD, node u will recalculate whether it needs rebroadcast the packet. This process is continued until either the RAD expires and the packet is then sent, or the packet is dropped (when all its neighbors are already covered by the broadcasts of some of its neighbors).

Lipman, Boustead and Judge [**?**] described the following broadcasting protocol. Upon receiving a broadcast message(s) from a node h, each node i (that was determined by h as a forwarding node) determines which of its one-hop neighbors also received the same message. For each of its remaining neighbors j (which did not receive a message yet, based on i's knowledge), node i determines whether j is closer to i than any one-hop neighbors of i (that are also forwarding nodes of h) who received the message already. If so, i is responsible for message transmission to j, otherwise it is not. Node i then determines a transmission range equal to that of the farthest neighbor it is responsible for.

III. Broadcasting Based on Sparse Topology Constructed Locally

In this section, we will study the energy efficient broadcasting based on some sparse structures constructed efficiently in a localized manner. Notice that the approximation of the minimum connected dominating set consumes power within a constant factor of the minimum when the power of each node is at some fixed value. On the other hand, majority of the spare structures consume power not much worse compared with the optimum when each node can adjust its power accordingly and the power needed to support a link uv is proportional to $\|uv\|^\beta$. However, none of these structures works well when the power needed to support a link uv is $c + \|uv\|^\beta$, where c is some fixed overhead of a node when processing and sending the signal. In addition, although the power consumption based on previous sparse structures is reasonable for random input, the average number of hops between all nodes and the source is large because these kind of structures prefer using short links to save the power consumption. As a tradeoff, A new structure by applying a spare structure (such as IMRG) on top of a hierarchical structure (such as a CDS) is proposed. For completeness of presentation, we first study each of them individually.

A. Distributed CDS

Recently, several algorithms were proposed with a constant worst case approximation ratio by taking advantage of the geometry properties of the underlying graph. Alzoubi *et al.* [16] gave the first fully localized algorithm to build a CDS which uses only $O(N)$ messages where N is the number of nodes. Alzoubi also gives a method to maintain mobility of nodes. The algorithm is as follows:

By definition, any pair of nodes in a MIS (Maximal Independent Set) are separated by at least two hops. However, a subset of nodes in a MIS U may be three hops away from its complementary subset in U. This case may appear when an ID-Based approach is used for rank assignment [1]kkk. Our distributed construction of the CDS can be briefly described as two phases. The first phase, a MIS S is constructed. The nodes in S are referred to as *dominators*, and the nodes not in S are referred to as *dominatees*. In the second phase each dominatee node identifies the dominators that are at most two hops away from itself and broadcasts this information. Using such information from all neighbors, each dominator node identifies a path of at most three hops (not necessarily the shortest one) to each dominator that is at most three hops away from itself and has larger ID than its own ID, and informs all nodes in this path about this selection. The set C then consists of all dominatee nodes in these paths, which are referred to as *connectors*. However, the description of our CDS construction combines the two phases. In the next, we describe a distributed algorithm with linear message complexity and linear time complexity to implement this distributed construction.

A.1 Local Variables and Structures

: Each node is in one of the four states: *candidate*, dominatee, dominator and connector. Each node is initially in the candidate state and subsequently enters either the dominatee state or the dominator state. The connector state can only be entered from the dominatee state. Each node also maintains several local variables and data structures. The local variable x_1 stores the number of current candidate neighbors, and is initially equal to the total number of neighbors. The local variable x_2 stores the number of current candidate neighbors with lower IDs, and is initially equal to the total number of neighbors with lower IDs. Note that both x_1 and x_2 can be initialized in linear time.

Each dominatee or connector node maintains a local variable y which counts the number of neighboring dominatees that have reported their list of adjacent dominators. y initially equals to 0. Each dominator node maintains a local variable z which counts the number of reports yet to be received from its neighbors on their lists of single-hop dominators and lists of two-hop dominators. z initially equals to *twice* the number of neighbors. Each dominatee node maintains two lists, $list1$ and $list2$. $list1$ stores the IDs of the neighboring dominators. Each entry in $list1$ is simply the ID of neighboring dominator. $list2$ stores the IDs of the dominators two hops away and the IDs of the neighboring dominatee to reach these dominators. Each entry in $list2$ is an ordered pair of the ID of a dominator two hops away and a neighboring dominatee that is adjacent to both. All entries in both lists are sorted in the increasing order of the IDs of the dominators, and both lists initially are empty. Each dominator node maintains two lists, $list2$ and $list3$. $list2$ (respectively, $list3$) stores the ID of the dominators with larger IDs that are two (respectively, three) hops away and the IDs of its neighbors to reach these dominators. An entry in $list2$ (respectively, $list3$) is an ordered pair of the IDs of a dominator with larger ID that is two (respectively, three) hops away and a neighbor to reach this dominator. All entries in both lists are sorted in the increasing order of the IDs of the dominators, and all lists initially are empty. Each connector node maintains a list *Rlist* which is initially empty. Each entry in *Rlist* contains two parameters. The first parameter is a pair of IDs of two dominators to which it maintains connectivity. The second parameter contains the ID of the associated connector that connects the two dominators in the first parameter, if the two dominators are three hop distance. If the two dominators are two hop distance, the value of the second parameter is assigned to Null. Each node further maintains a list *Clist* which is initially empty and stores the IDs of neighboring connectors.

A.2 Messages and Actions

: A candidate node with $x_2 = 0$ changes its own state to dominator, initializes z to twice the number of neighbors, and then broadcasts a DOMINATOR message. Note that such node does exist at the beginning. Upon receiving a DOMINATOR message, a node (which cannot be a dominator node) decrements x_1 by one and inserts the ID of the sender into $list1$. A candidate node further proceeds as follows. It changes its own state to dominatee, and then broadcasts a DOMINATEE message. If $x_1 = 0$ after the updating, it broadcasts a LIST1 message which contains all entries in $list1$; if the number of neighboring dominators is also equal to the number of neighbors (i.e., all neighbors are dominators), it also broadcasts a LIST2 message which contains all entries in $list2$ (which is empty in this case).

Upon receiving a DOMINATEE message, a candidate node decrements x_1 by one. If the sender has lower ID, it decrements x_2 by one. If $x_2 = 0$ after the updating, it first changes its own state to dominator, then initializes z to twice the number of neighbors, and finally broadcasts a DOMINATOR message. Upon receiving a DOMINATEE message, a dominatee node decrements x_1 by one. If $x_1 = 0$ after the updating, it broadcasts a LIST1 message which contains all entries in $list1$. Upon receiving a LIST1 message, a dominatee or connector node increments y by one. (When a node receives a LIST1 message, the node cannot be in candidate state. However, some of its neighbors may be still in the candidate state and thus it can not determine the final number of neighboring dominatees. This is why we increment y.) For each dominator ID contained in the LIST1 message which does not appear in the current $list1$ and $list2$, it inserts into $list2$ an entry consisting of this dominator ID and the senders ID. Finally, if $x_1 = 0$ and y is also equal to the number of neighbors minus the number of neighboring dominators (the length of $list1$) after the updating, it broadcasts a LIST2 message which contains all entries in $list2$. Upon receiving a LIST1 message, a dominator node decrements z by one. For each dominator ID contained in the LIST1 message which is larger than its own ID and does not appear in the current $list2$, it inserts into $list2$ an entry consisting of this dominator ID and the senders ID, and removes from $list3$ the entry containing this dominator ID if there is any. If $z = 0$ after the updating, it broadcasts a LIST3 message which contains all entries in $list2$ and $list3$. Upon receiving a LIST2 message, a dominator node decrements z by one. For each dominator ID contained in the LIST2 message which is larger than its own ID and does not appear in the current $list2$ and $list3$, it inserts into $list3$ an entry consisting of this dominator ID and the senders ID. If $z = 0$ after the updating, it broadcasts a LIST3 message which contains all entries in $list2$ and $list3$. Upon receiving a LIST3 message, a node (which must be either in dominatee state or in connector state) checks whether its ID appears in any of the entries in this message, and if so it proceeds as follows. First, it sets its state to connector if its current state is dominatee. Then, for each entry in LIST3 message that has its ID, it inserts into the first parameter of its *Rlist* the ID of the sender, and the ID of the dominator it is responsible to connect (target dominator). If the target dominator is adjacent to itself, it sets the second parameter to null. Otherwise, it sets the second parameter to the ID of neighboring node associated with the target dominator in its own $list2$. Finally, it broadcasts a CONNECTOR1 message which includes two parameters, the first parameter has its own ID, and the second parameter contains a list of all entries that were added to its *Rlist*. Upon receiving a CONNECTOR1 message, a node inserts into its *Clist* the ID of the sender. A node which is not a dominator further checks whether its ID appears in any of the entries of the second parameter of the message, and if so it proceeds as follows. First, it sets its state to connector if its current state is dominatee. Then, it inserts into the first parameter of its *Rlist* the first parameter of the entry that has its ID in the received CONNECTOR1 message, and adds the ID of the sender to the second parameter in its *Rlist*. Finally, it broadcasts a CONNECTOR2 message. Upon receiving a CONNECTOR2 message, a node inserts into its *Clist* the ID of the sender. Figure 3 illustrates the construction process of the CDS. In the graph, the IDs of the nodes are labelled beside the nodes. White nodes represent the candidate nodes, black nodes represent the dominators, gray nodes represent the dominatees, and the white node with an inner black node represents a connector node. A possible execution scenario is shown in Figure 3(a)(d), which is explained below.

1. Initially all nodes are candidates (see Figure 3(a)).
2. Each of the nodes 1, 2, 3 and 4 declares itself as a dominator, and broadcasts a DOMINATOR message. Notice this process may occur simultaneously, since each one of these nodes has the lowest ID among all its neighbors. Whenever a neighboring node receives the DOMINATOR message, it declares itself as a dominatee and broadcasts a DOMINATEE message. Thus each of the nodes 5, 6 and 7 declares itself as a DOMINATEE and broadcasts a DOMINATEE message. (see Figure 3(b)).
3. Upon receiving DOMINATOR and DOMINATEE messages from all its neighbors; node 5 sends a LIST1 message, which includes the IDs of nodes 1 and 2; node 6 sends a LIST1 message, which includes the IDs of nodes 3 and 4; and node 7 sends a LIST1 message, which includes the IDs of nodes 3 and 4.
4. Upon receiving the LIST1 message from node 5, node 6 sends a LIST2 message, which includes the IDs of nodes 1 and 2. Upon receiving the LIST1 message from node 6, node 5 sends a LIST2 message, which includes the IDs of nodes 3 and 4. Since all neighbors of node 7 are dominators, node 7 sends a LIST2 message with the empty list $list2$.
5. Upon receiving LIST1 and LIST2 messages from node 5, node 1 selects node 5 as a connector to reach nodes 2, 3 and 4 by sending a LIST3 message. Upon receiving LIST1 and LIST2 messages from node 5, node 2 selects node 5 as a connector to reach nodes 3 and 4 by sending a LIST3 message. Upon receiving LIST1 and LIST2 messages from nodes 6 and 7, node 3 selects node 7 as a connector to reach node 4 by sending a LIST3 message. Notice, node 4 does not make any selection since it has the largest ID among all dominators within three-hop distance.
6. Upon receiving LIST3 message from nodes 1 and 2, node 5 declares itself as a connector for each of the pairs $(1, 2), (1, 3), (1, 4), (2, 3)$ and $(2, 4)$, then it sends a CONNECTOR1 message selecting node 6 as a second connector to connect each of the nodes 1 and 2 to each of the nodes 3 and 4. Upon receiving LIST3 message from node 3, node 7 declares itself as a connector for the pair $(3, 4)$, then it sends a CONNECTOR1 message. (see Figure 3(c)).
7. Upon receiving the CONNECTOR1 message from node 5, nodes 6 declares itself as a connector for each of the pairs $(1, 3), (1, 4), (2, 3)$ and $(2, 4)$ and it broadcasts CONNECTOR2 message.(see Figure 3(d)).

A.3 Message and Time Complexity

Theorem 1: Alzoubi's distributed algorithm for constructing a CDS has an $O(n)$ time complexity, and $O(n)$ message complexity. [16]

A.4 Mobile maintenance

We need to maintain a connected dominating set in the unit-disk graph as the topology of the network may change. In the mean time we need to maintain the same performance ratio for the CDS. The key technique in our approach is to maintain the MIS in the unit disk graph first, and to maintain the connection between all MIS nodes within three-hop distance through connector nodes. In our discussion for the maintenance of the CDS, we need to distinguish between dominators and connectors. After any topology changes, the MIS should be maintained, but there may be an additional affect on the connectors. When a dominator node is turned off, or leaves its vicinity, this changes should affect the connectors, which are used only to connect this dominator to other dominators. After the MIS is maintained and the connectors are changed back to dominatees whenever is needed, the next step is to make sure that any dominator appears in a new vicinity must have a two-hop and three-hop path of connector nodes to all two-hop and three-hop dominators respectively. In the next, we provide a brief description of the maintenance process. The implementation details of this process will appear in Alzoubis Dissertation.

Dominator Node Movement: When a dominatee or connector node v learns that a dominator node u has left its vicinity and u is the only dominator of v, v changes its own state to candidate and then it sends a WARNING1 message reporting the loss of u. The WARNING1 message contains vs ID and state, and the ID of u. If v has other dominators and v is a dominatee, it simply remains as a dominatee. If v is a connector connecting two or more dominators other than u, it remains as a connector. Otherwise, it changes its state to dominatee, and sends an SDOMINATEE message. Whenever a dominatee node w receives a WARNING1 message from v reporting the loss of u, it sends a RESPONSE, which contains ws ID and state, and the ID of u. Whenever a connector node w receives a WARNING1 message from a dominatee node v, or from a connector node v which is not in ws *Rlist*, w maintains its state as a connector, and sends a RESPONSE message. Whenever a connector node w receives a WARNING1 message from a connector node v, and w is only responsible to connect u to other dominators, w changes its state to dominatee and sends a RESPONSE message. Otherwise, w maintains its state as a connector, and sends a RESPONSE message. Whenever a connector

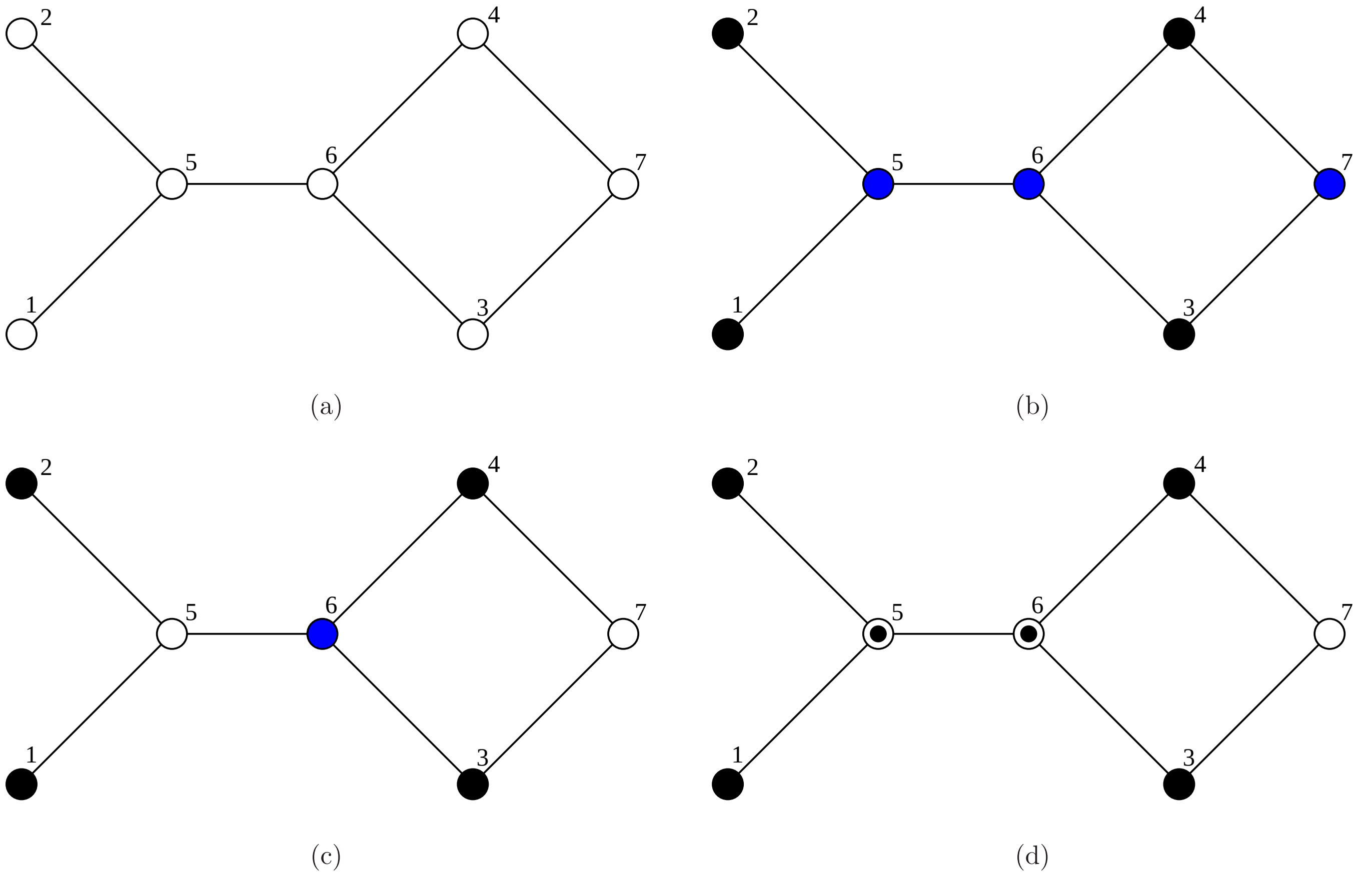


Fig. 3. CDS construction example.

node w receives an SDOMINATEE message from a connector node v, and w is only responsible to connect u to other dominators, w changes its state to dominatee and sends an SDOMINATEE message. Otherwise, w maintains its state as a connector. Whenever candidate node receives a WARNING1, or RESPONSE message from each neighbor, it applies the CDS algorithm locally. Thus, a candidate node v with the lowest ID declares itself as a dominator. Then v must be connected through connector nodes to all dominators within three-hop distance by applying the CDS algorithm locally. When a dominator node u joins a neighborhood with at least one dominator, the dominator with the lowest ID becomes a winner, and maintains its state as a dominator. All other dominators switch their state to a dominatee. Otherwise, u (winner) maintains its state as a dominator, and sends a DOMINATOR message. However, the winner must be connected through connector nodes to all dominators within three-hop distance by applying the CDS algorithm locally.

Dominatee or Candidate Node Movement: When a dominatee node v joins a new neighborhood, if any of its new neighbors is a dominator, it maintains its state as a dominatee, it also sends a LIST1 message. When receiving a LIST1 message from v, a dominatee or connector node w sends a LIST1 and LIST2 messages. Whenever v receives a LIST1 message from each dominatee and connector neighbor, it sends a LIST2 message. Then the dominators receiving the LIST1 and LIST2 messages react based on the CDS algorithm. When a dominatee node u joins a new neighborhood, if non of its new neighbors is a dominator, it declares itself as a dominator and sends a DOMINATOR message. If a DOMINATOR message is received from y, a dominatee or connector node v sends a LIST1 message, followed by LIST2 message. When receiving a LIST1 message from a dominatee or connector neighbor v, a dominatee or connector node w sends a LIST2 message. Then the dominators receiving the LIST1 and LIST2 messages react based on the CDS algorithm. Whenever a new node y joins the network, initially it is a candidate, if any of its neighbors is a dominator, it becomes a dominatee, and the same action is taken as if a dominatee node joins a new neighborhood. Whenever a new node y joins the network, initially it is a candidate, if non of its neighbors is a dominator, it declares itself as a dominator, and sends a DOMINATOR message. Then the same action is taken as if a dominatee node joins a new neighborhood and becomes a dominator.

Connector Node Movement Whenever a connector node w learns that a connector node v has left its vicinity, if w is only responsible to connect v to other dominators, it changes its own state to a dominatee. Otherwise, it maintains its own state as a connector. However, w sends a LOST message reporting the lose of connection to the dominators (lost dominators) associated with it through the connector node v. Whenever a dominator node x receives a LOST message from w, if any of the lost dominators has a larger ID than its own, it sends a REQUEST message, which contains its own ID, and the IDs of all lost dominators with larger IDs. Whenever a dominatee or a connector node x receives the REQUEST message, it sends a REPLY message, which contains its own ID and for each dominator appeared in the REQUEST message the ordered pair (ID, distance), where distance is, equal to 1 if the dominator is one-hop from x, equal to 2 if the dominator is two-hop from x, or equal to 8 otherwise. Whenever a dominator x receives a REPLY message from a neighbor, it selects new connectors for all two-hop and three-hop dominators, then the CDS continues to be applied locally. Whenever a dominator node u learns that a connector node v has left its vicinity, if any of the dominators (lost dominators) connected to u through v has a larger ID than its own, it sends a REQUEST message, which contains its own ID, and the IDs of all lost dominators with larger IDs. Whenever a dominatee or a connector node x receives the REQUEST message, it sends a REPLY message, which contains its own ID and for each dominator appeared in the REQUEST message the ordered pair (ID, distance), where distance is, equal to 1 if the dominator is one-hop from x, equal to 2 if the dominator is two-hop from x, or equal to 8 otherwise. Whenever a dominator x receives a REPLY message from a neighbor, it selects new connectors for all two-hop and three-hop dominators, then the CDS continues to be applied locally.

Examples Figure 4(a,b) illustrates the action taking by neighboring nodes in response to a dominator node movement. Figure 4(a) represents the network topology before the node movement. When node 4 moves and becomes within the vicinity of the dominator node 3, and since it has a higher ID than node 4, it changes its state to a dominatee and sends an SDOMINATEE message. When node 7 receives the SDOMINATEE message from node 4, it removes each pair in its *Rlist* associated with the node 4. Since node 7 has only one entry in its *Rlist*, and this entry corresponds to node 4, node 7 switches to dominatee and sends an SDOMINATEE message.

Figure 5(a-e) illustrates the action taking by neighboring nodes in response to a dominator node movement and two broken links simultaneously. Figure 5(a) represents the network topology before the node movement. The execution procedures are explained below:

1. When the dominator node 1 moves upward, both of the dominatee nodes 4 and 5 become candidate nodes (see 5(b)), and both of them send a WARNING message.
2. Whenever node 2 receives the WARNING message from node 5, it sends back a RESPONSE message. Whenever node 4 receives the WARNING message from node 5, it declares itself as a dominator (since it has the lowest degree among all its candidate neighbors) (see 5(c)) and sends a DOMINATOR message.
3. Whenever node 5 receives the DOMINATOR message, it declares itself as a dominatee (see 5(d)) and sends a DOMINATEE message, followed by LIST1 and LIST2 messages.
4. Whenever node 2 receives the LIST1 message it sends a LIST2 message.
5. Whenever the dominator node 3 receives the LIST2 message from node 2, it sends a LIST3 message selecting node 2 as a connector to reach the dominator node 4 (since it has a lower ID than the dominator node 4).
6. Whenever node 2 receives the LIST3 message, it selects node 5 as a second connector to reach node 4, by sending a CONNECTOR1 message.
7. Whenever node 5 receives the CONNECTOR1 message addressed to itself, it declares itself as a connector (see 5(e)) and sends a CONNECTOR2 message.

Recently, Wan, *et al.* [45] proposed a communication efficient algorithm to find connectors based on the fact that there are only a constant number of dominators within k-hops of any node. The following observation is a basis of several algorithms for CDS. After clustering, one dominator node can be connected to many dominatees. However, it is well-known that a dominatee node can only be connected to at most *five* dominators in the unit disk graph model.

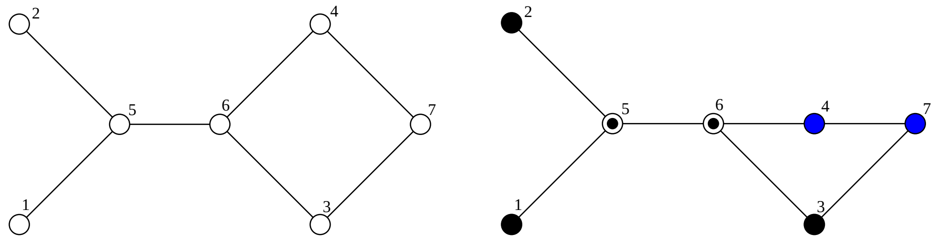
Generally, it was shown in [45], [46] that for each node (dominator or dominatee), there are at most a constant $\ell_k < (2k+1)^2$ number of dominators that are at most k hops away.

Given a dominating set S, let $VirtG$ be the graph connecting all pairs of dominators u and v if there is a path in UDG connecting them with at most 3 hops. $VirtG$ is connected. It is natural to form a connected dominating set by finding connectors to connect any pair of dominators u and v if they are connected in $VirtG$. This strategy is also adopted by Wan, *et al.* [45]. Wan *et al.* [16] suggested to find only one unique shortest path to connect any two dominators that are at most three hops away. Wang and Li [46] and Alzoubi *et al.* [45] discussed in detail some approaches to optimize the communication cost and the memory cost. In [16], [46], they proved the following theorem.

Theorem 2: The number of connectors found by this algorithm is at most ℓ_3 times of the minimum. The size of the connected dominating set found by this algorithm is within a small constant factor of the minimum.

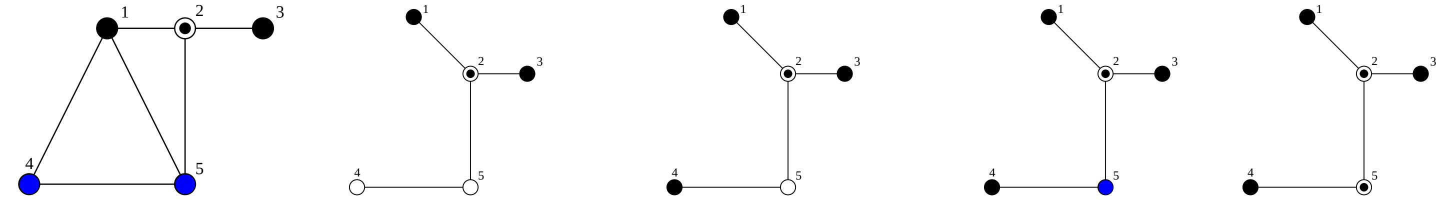
The graph constructed by this algorithm is called a CDS graph (or *backbone* of the network). If we also add all edges that connect all dominatees to their dominators, the graph is called extended CDS, denoted by CDS'. It was shown in [16], [46] that the CDS' graph is a sparse spanner in terms of both hops and length with factors 3 and 6, meanwhile CDS has a bounded node degree $\max(\ell_3, 5 + \ell_2)$. See [46] for detailed proofs.

Several routing algorithms require the underlying topology be planar. Notice in the formation algorithm of CDS, we do not use any geometry information. The resulting CDS maybe non-planar graph. Even using some geometry information, the CDS still is not guaranteed to be a planar graph. Then Li *et al.* [46] proposed a method to make the graph CDS planar without losing the spanner property of the backbone. Their method applies the localized Delaunay triangulation [47] on top of the induced graph from



(a) (b)

Fig. 4. Node movement example.



(a) (b) (c) (d) (e)

Fig. 5. Dominator node movement example with two simultaneous broken links.

CDS, denoted by ICDS. It was proved in [47] that $LDel(G)$ is a spanner if G is a unit disk graph. Notice that ICDS is a unit disk graph defined over all dominators and connectors. Consequently, $LDel(ICDS)$ is a spanner in terms of length.

B. RNG and Variants

Although RNG is very sparse structure (the average number of neighbors per node is about 2.5), in some degenerate cases a particular node may have arbitrarily large degree. This motivated Stojmenovic [48] to define a modified structure where each node will have degree bounded by 6. The same structure was independently proposed by Li in [41], with an additional motivation. Li proved that the modified RNG is the first localized method to construct a structure H with weight $O(\omega(MST))$ using total $O(n)$ local-broadcast messages. Note that, if each node already knows the positions and IDs of all its neighbors, then no messages are needed to decide which of its edges belong to (modified) RNG. Notice that, traditionally, the relative neighborhood graph will always select an edge uv even if there is some node on the boundary of $lune(u, v)$. Here $lune(u, v)$ is the intersection of two disks centered at nodes u and v with radius $\|uv\|$ respectively. Thus, RNG may have unbounded node degree, e.g., considering $n-1$ points equally distributed on the circle centered at the nth point v, the degree of v is $n-1$. Notice that for the sake of lowering the weight of a structure, the structure should contain as less edges as possible without breaking the connectivity. Li [41] and Stojmenovic [48] then naturally extended the traditional definition of RNG as follows.

We need to make distinct edge lengths. This can be achieved by adding the secondary, and if necessary, the ternary keys for comparing two edges. Each node is assumed to have a unique ID. Then consider the record $(\|uv\|), ID(u), ID(v))$, where $ID(u) < ID(v)$ (otherwise u and v are exchanged for given edge). Two edges compare their lengths first to decide which one is longer. If same, they then compare their secondary key, which is their respective lower endpoint node's ID. If this is also same, then the ternary key resolves the comparison (otherwise we are comparing edge against itself). This simple method for making distinct edge length was proposed in [49], [40]. The edge lengths, so defined, are then used in the regular definition of RNG. It is easy to show that two RNG edges uv and uw going out of the same node must have angle between them at least $\pi/3$, otherwise $vw < uv$ or $vw < uw$, and one of the two edges becomes the longest in the triangle and consequently could not be in RNG. Li [41] denoted modified RNG structure by RNG'. Obviously, RNG' is a subgraph of traditional RNG. It was proven in [41], [48] that RNG' still contains a MST as a subgraph. However, RNG' is still not a low weight structure. We then study some structures proven to be low weight.

C. Localized Low Weight Structures

C.1 Sparse and Low Weight Structure H

Notice that it is well-known that the communication complexity of constructing a minimum spanning tree of a n-vertex graph G with m edges is $O(m+n\log n)$; while the communication complexity of constructing MST for UDG is $O(n\log n)$ even under the local broadcasting communication model in wireless networks. It was shown in [41] that it is *impossible* to construct a low-weighted structure using only one hop neighbor information.

The localized algorithm given in [41] that constructs a low-weighted structure using only some two hops information is as follows.

Algorithm 1: Construct Low Weight Structure H

1. All nodes together construct the graph RNG' in a localized manner.
2. Each node u locally broadcasts its incident edges in RNG' to its one-hop neighbors. Node u listens to the messages from its one-hop neighbors.
3. Assume node u received a message informing existence of edge $xy \in RNG'$ from its neighbor x. For each edge $uv \in RNG'$, if uv is the longest among uv, xy, ux, and vy, node u removes edge uv. Ties are broken by the label of the edges. Here assume that $uvyx$ is the convex hull of u, v, x, and y.
4. Let H be the final structure formed by all remaining edges in RNG'.

Obviously, if an edge uv is kept by node u, then it is also kept by node v. The following theorem was proved in [41].

Theorem 3: [41] The total edge weight of H is within a constant factor of that of the minimum spanning tree.

This was proved by showing that the edges in H satisfy the *isolation property* (defined in [50]). They [41] also showed that the final structure contains MST of UDG as a subgraph.

Clearly, the communication cost of Algorithm 1 is at most $7n$: initially each node spends one message to tell its one-hop neighbors its position information, then each node uv tells its one-hop neighbors all its incident edges $uv \in RNG'$ (there are at most total $6n$ such messages since RNG' has at most $3n$ edges). The computational cost of Algorithm 1 could be high since for each link $uv \in RNG'$, node u has to test whether there is an edge $xy \in RNG'$ and $x \in N_1(u)$ such that uv is the longest among uv, xy, ux, and vy. Then [42] presents some new algorithms that improve the computational complexity of each node while still maintains low communication costs.

C.2 Spare and Low Weight Structure LMST$_k$

The first new method in [42] uses a structure called *local minimum spanning tree*, let us first review its definition. It is first proposed by Li, Hou and Sha [40]. Each node u first collects its one-hop neighbors $N_1(u)$. Node u then computes the minimum spanning tree $MST(N_1(u))$ of the induced unit disk graph on its one-hop neighbors $N_1(u)$. Node u keeps a directed edge uv if and only if uv is an edge in $MST(N_1(u))$. They call the union of all directed edges of all nodes the *local minimum spanning tree*, denoted by $LMST_1$. If only symmetric edges are kept, then the graph is called $LMST_1^-$, i.e., it has an edge uv iff both directed edge uv and directed edge vu exist. If ignoring the directions of the edges in $LMST_1$, they call the graph $LMST_1^+$, i.e., it has an edge uv iff either directed edge uv or directed edge vu exists. They prove that the graph is connected, and has bounded degree 6. In [42], Li *et al.* also showed that graph $LMST_1^-$ and $LMST_1^+$ are actually planar. Then they extend the definition to k-hop neighbors, the union of all edges of all minimum spanning tree $MST(N_k(u))$ is the k *local minimum spanning tree*, denoted by $LMST_k$. For example, the 2 local minimum spanning tree can be constructed by the following algorithm.

Algorithm 2: Construct Low Weight Structure $LMST_2$ by 2-hop Neighbors
1. Each node u collects its two hop neighbors information $N_2(u)$ using a communication efficient protocol described in [44].
2. Each node u computes the Euclidean minimum spanning tree $MST(N_2(u))$ of all nodes $N_2(u)$, including u itself.
3. For each edge $uv \in MST(N_2(u))$, node u tells node v about this directed edge.
4. Node u keeps an edge uv if $uv \in MST(N_2(u))$ or $vu \in MST(N_2(v))$. Let $LMST_2^+$ be the final structure formed by all edges kept. It keeps an edge if either node u or node v wants to keep it. Another option is to keep an edge only if both nodes want to keep it. Let $LMST_2^-$ be the structure formed by such edges.

In [42], they prove that structures $LMST_2$ ($LMST_2^+$ and $LMST_2^-$) are connected, planar, low-weighted, and have bounded node degree at most 6. In addition, MST is a subgraph of $LMST_k$ and $LMST_k \subseteq RNG'$. Although the constructed structure $LMST_2$ has several nice properties such as being bounded degree, planar, and low-weighted, the communication cost of Algorithm 2 could be very large to save the computational cost of each node. The large communication costs are from collecting the two hop neighbors information $N_2(u)$ for each node u. Although the total communication of the protocol described in [44] is $O(n)$, the hidden constant is large.

C.3 Spare and Low Weight Structure IMRG

A method was presented in [42] to improve the communication cost of collecting $N_2(u)$ by using a subset of two hop information without sacrificing any properties. Define $N_2^{RNG'}(u) = \{w \mid vw \in RNG' \text{ and } v \in N_1(u)\} \cup N_1(u)$. The modified algorithm is as follows.

Algorithm 3: Construct Low Weight Structure $IMRG$ by 2-hop Neighbors in RNG'
1. Each node u tells its position information to its one-hop neighbors $N_1(u)$ using a local broadcast model. All nodes together construct the graph RNG' in a localized manner.
2. Each node u locally broadcasts its incident edges in RNG' to its one-hop neighbors. Node u listens to the messages from its one-hop neighbors.
3. Each node u computes the Euclidean minimum spanning tree $MST(N_2^{RNG'}(u))$ of all nodes $N_2^{RNG'}(u)$, including u itself.
4. For each edge $uv \in MST(N_2^{RNG'}(u))$, node u tells node v about this directed edge.
5. Node u keeps an edge uv if $uv \in MST(N_2^{RNG'}(u))$ or $vu \in MST(N_2^{RNG'}(v))$. Let $IMRG^+$ be the final structure formed by all edges kept. Similarly, the final structure is called $IMRG^-$ when edge $uv \in RNG'$ is kept iff $uv \in MST(N_2^{RNG'}(u))$ and $uv \in MST(N_2^{RNG'}(v))$. Here IMRG is the abbreviation of *Incident MST and RNG Graph*.

Notice that in the algorithm, node u constructs the local minimum spanning tree $MST(N_2^{RNG'}(u))$ based on the induced UDG of the point sets $N_2^{RNG'}(u)$. It is obvious that the communication cost of Algorithm 3 is at most $7n$.

It is shown that structures $IMRG^+$ and $IMRG^-$ are still connected, planar, bounded degree, and low-weighted. They are obviously planar, and with bounded degree since both structures are still subgraphs of the modified relative neighborhood graph RNG'. Clearly, the constructed structures are supergraphs of the previous structures, i.e., $LSMT_2+ \subseteq IMRG^+$ and $LSMT_2^- \subseteq IMRG^-$, since Algorithm 3 uses less information than Algorithm 2 in constructing the local minimum spanning tree. Both $IMRG^-$ and $IMRG^+$ have node degree at most 6.

Recall that until now there is no efficient localized algorithm that can achieve all following desirable features: bounded degree, planar, low weight and spanner. It is still an open problem.

D. Combining Clustering and Low Weight

We then discuss in detail a new approach by combining the low-weighted structures and the connected dominating set for energy efficient broadcasting in traditional one-to-many (omnidirectional antenna) networks. Combining structures for efficient broadcasting is not new. Seddigh, Gonzalez and Stojmenovic [51] specified two location based broadcasting algorithms that combine RNG and internal node concept (connected dominating set) as follows. PI-broadcast algorithm applies the planar subgraph construction first, and then applies the internal nodes concept on the subgraph. The result is different from the internal nodes applied on the whole graph. IP-broadcast algorithm changes the order of concept application compared to the previous algorithm. Internal nodes are first identified in the whole graph, and then the obtained subgraph (containing only internal nodes) is further reduced to planar one by the RNG construction. We found that using IMRG on top of CDS provides us some extra savings in terms of total power consumption.

Notice that the constructed low-weighted structures are subgraphs of RNG, thus, they are sparser than RNG and thus could save energy, which is also validated in our simulations. Here, we concentrate on *IP-broadcast*. We first construct a CDS and then apply the IMRG structure on top of the CDS to remove some long links. Local improvement can also be applied after the structure is constructed. However, we did not implement them since the effect of the local improvement may hinder our ability to study what causes the performance improvement of the new broadcasting method.

E. A Negative Result

In [41], [42], they showed that it is impossible to design a deterministic localized method that constructs a structure such that the broadcasting based on this structure consumes energy within a constant factor of the optimum when the power needed to support a link uv is $\|uv\|^\beta$. Assume that there is a deterministic localized algorithm to do so: it uses k-hop information of every node u to select the edges incident on u, and the energy consumption is no more than C times of the optimum. They construct two set of nodes configurations such that the k-hop information collected in a special node u is same for both configurations. In addition, there is an edge uv in both UDGs such that if node u decides to keep edge uv (then edge uv is kept in both configurations), the energy consumption of one configuration is already more than C times of the optimum; if node u decides to remove edge uv (then edge uv is removed in both configurations), then the structure constructed for another configuration is disconnected.

IV. Experimental studies

We conducted extensive simulations to study the performance of different methods that are based on virtual backbone and spare structures. We model the network by unit disk graph and mobile hosts are randomly placed in a two dimensional area. We tried unit disk graph with 1000 nodes that are randomly placed in squares of different sizes, squares of size 5×5 to 15×15 and the transmission range is fixed to one unit in all simulations. For each square we tried 100 different graphs and we measured the following metrics:

1. **Number of messages:** Since wireless nodes are often powered by batteries only and have limited memories, wireless ad hoc networks prefer localized and power efficient algorithms. Message sending consumes energy and for the reasons mentioned above, the number of messages would be a very important metric in all wireless ad hoc network algorithms. We would like to compare the message complexity of constructing different structures.
2. **Adjustable Power:** After constructing the structure for broadcasting, we assign each leaf node to its closest non-leaf neighbor to keep the network connected. Each non-leaf node needs as much energy as it could reach its farthest neighbor in the broadcasting structure.
3. **Non-adjustable Power:** After constructing the structure for broadcasting, we assign each non-leaf node its maximum power, which is r^β where r is the transmission range. In this case, the total power of the network is proportional to the number of non-leaf nodes. Thus, having less number of non-leaf nodes results in less total power consumption for broadcasting.

We compare different structures constructed either in a centralized manner or in a localized manner for minimum energy broadcasting. The structures we studied are MST, BIP, CDS, RNG', H, LMST_k, IMRG, IMRG-CDS. Figure 6 illustrates all such structures constructed for 500 nodes distributed in a 5×5 square and the transmission range of all nodes is one unit.

A. Number of messages

One important metric when comparing different locally constructed structure is the number of messages used. LMST_1 and RNG need only the information of one hop neighbor, so the total message needed to construct them is n, where n is the number of wireless nodes. We already know that IMRG can be constructed using at most $7n$ messages and our simulations validate that. Although constructing CDS uses $O(n)$ messages but the hidden constant could be large. Figure 7 compares the number of mes-

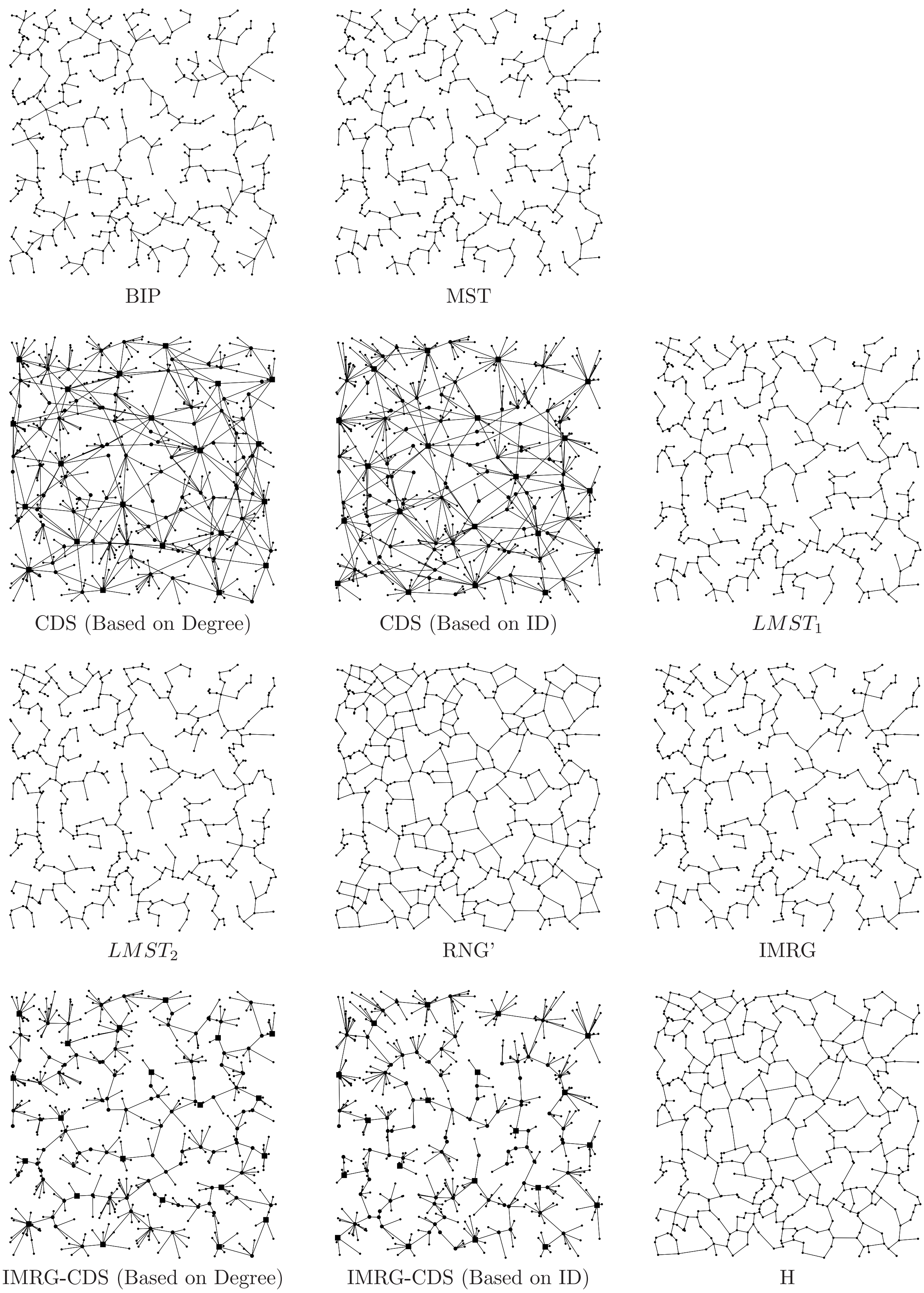


Fig. 6. Comparison between different power models.

sages used in constructing different structures locally. As can be seen in Figure 7 CDS construction uses lots of messages when the graph is dense and most of these messages are used during the phase of finding connectors to connect dominators that are three-hops away from each other. Notice that the structure IMRG-CDS is built upon CDS, thus it always uses more messages than the CDS. Figure 7 shows that the difference between the number of messages used for *CDS* and IMRG-CDS is not high. In other words, most of the messages used in building IMRG-CDS are used in building the CDS.

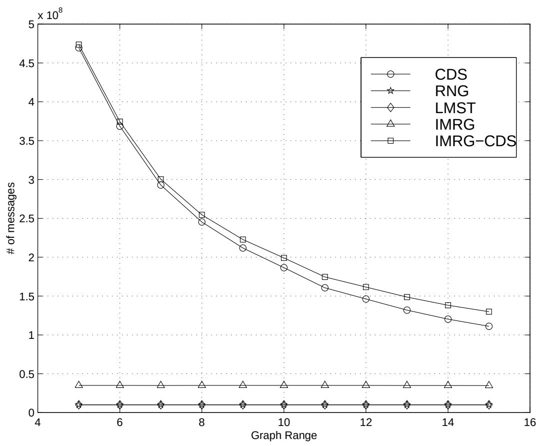


Fig. 7. Number of messages used in constructing structures locally.

B. Adjustable power

In Section II, we discussed two power models for node: adjustable and non-adjustable. In adjustable power model, the structures that have smaller edges perform better than the structures that have longer edges. Figure 8 compares the total power consumption of the broadcasting based on different localized structures when adjustable power model is used. Since CDS tries to find the backbone with minimum cardinality, it has lots of long edges, i.e., edges whose length is close to the maximum transmission range. Not surprisingly, CDS performs poorly in adjustable power model in our simulations. Figure 8 shows that the power consumption of broadcasting based on other topologies are only slightly different, (IMRG is always the best and RNG is always the worst).

We further compared the structures constructed locally with the structure MST and BIP constructed in a centralized way under the adjustable power model and plotted the results in Figure 9. Although the centralized methods have better performance but the difference between localized and centralized methods is not dramatic. Interestingly, we found that the difference becomes larger when the network becomes sparser (in our simulations, it means that the geometry region where the 1000 nodes with transmission range 1 reside becomes bigger). For example, when the geometry region is a 15 by 15 square, IMRG only consumes power about 33% more than MST, while RNG consumes about 96%, and LMST consumes about 48% percent more power than MST. Notice that IMRG-CDS consumes about

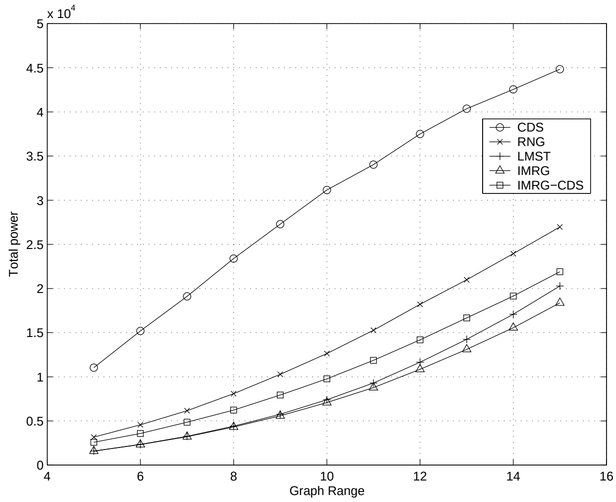


Fig. 8. Adjustable power model.

55% more power than MST, but the structure IMRG-CDS has smaller number of hops to connect the source to most of the destinations.

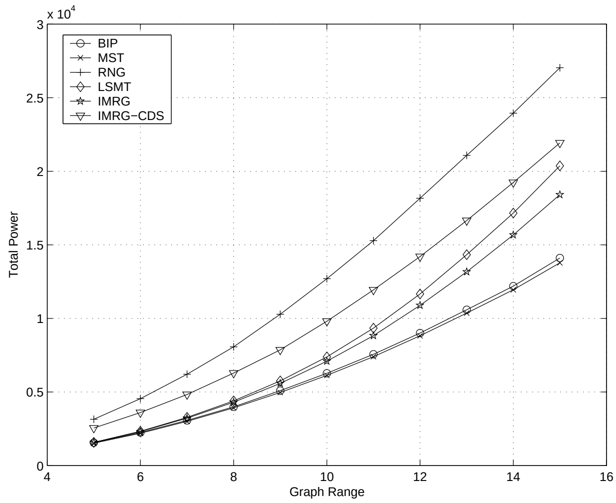


Fig. 9. Adjustable power model.

C. Non-adjustable power

Our first simulations confirmed that the sparse structures perform well for broadcasting for adjustable power model where the power needed for a link uv is $\|uv\|^{\beta}$. We continue to study their performance under the non-adjustable power model. In non-adjustable power model, all non-leaf nodes in the structure use the maximum power. Thus, the less number of non-leaf nodes the structure has, the less power it consumes. Our simulations show that CDS and IMRG-CDS have less number of non-leaf nodes than all other structures discussed in this paper. Consequently, these two structures perform much better in non-adjustable power model. Figure 10 compares the total power consumption of different structure used for broadcasting. Interestingly, we found that the difference becomes larger when the network becomes denser (in our simulations, it means that the geometry region where the 1000 nodes with transmission range 1 reside becomes smaller). To be consistent with the adjustable power case, for example, when the geometry region is a 15 by 15 square, IMRG-CDS only consumes power about 96% of the power consumed by CDS, while RNG consumes about 86%, and LMST consumes about 70% percent more power than CDS. RNG consumes power about 6 times the power consumed by CDS when the geometry region is a 5 by 5 square. IMRG-CDS always consumes the least power for non-adjustable power model here.

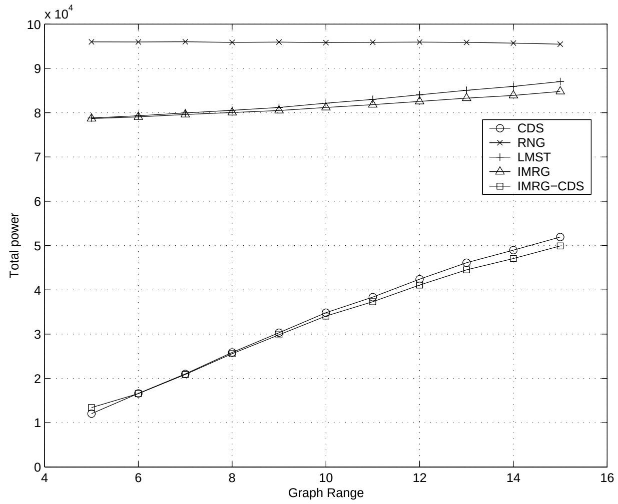


Fig. 10. Non-adjustable power model.

We further compared the structures constructed locally and structures constructed in a centralized manner and plotted the results in Figure 11. Since BIP and MST are based on adjustable power model, they perform poorly in non-adjustable power model. Notice that IMRG performs the worst when the node power is fixed because it has the largest number of non-leaf nodes.

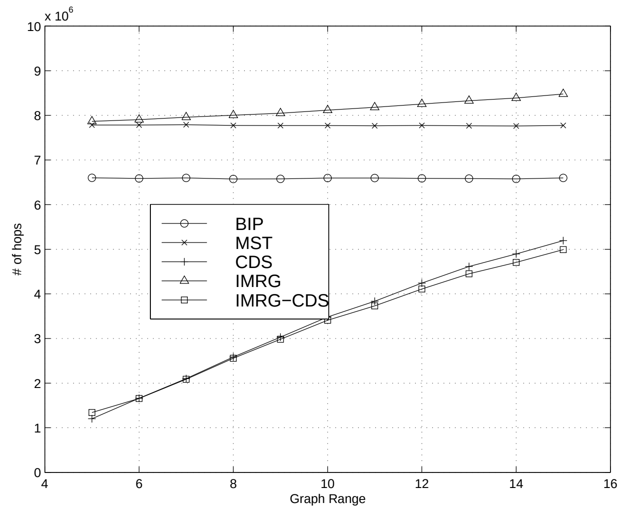


Fig. 11. Non-adjustable power model.

D. Using More Practical Energy Model

In all adjustable power models we assumed that the pass loss is the major part of power consumption to transmit signals. In other words we assumed that the power to support a link uv is $\|uv\|^{\beta}$, where $\beta \in [2, 5]$. In this section we take into account the constant value c and we study the performance of different methods based on power adjustable model where the power to support a link uv is $c + \|uv\|^{\beta}$.

When the value of c is considered, the power used for broadcasting increases by $c \times N_i$, where N_i is the number of non-leaf nodes in the structure. Thus, the structures that have less number of non-leaf nodes could perform better in this energy model. Figure 12 compares the power consumption for broadcasting of different structures when $c = 0$. As can be seen CDS performs poorly and IMRG-CDS performs slightly better than H and RNG and slightly worse than $IMRG$ and MST.

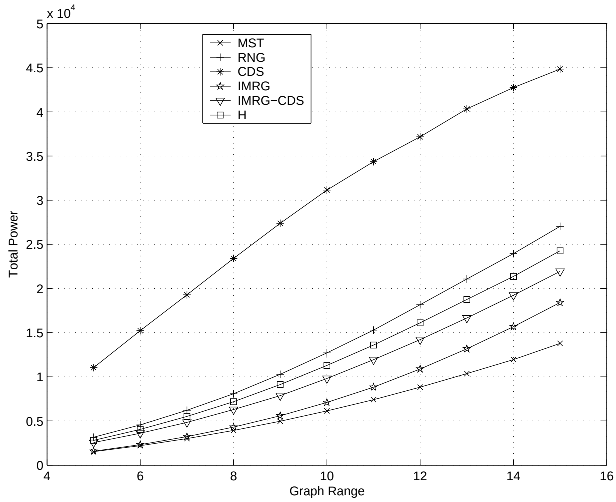


Fig. 12. Power model $c + \|uv\|^{\beta}$ when $c = 0$.

As the value of c increases,(See Figure 13 for example), IMRG-CDS becomes the best and CDS performs much better compared to the previous one. That is because of the following observation: in the structures that are designed for adjustable power we have lots of non-leaf nodes. Since the additional power consumed for broadcasting when c is not ignored is proportional to the number of non-leaf nodes, when the value of c increases, the increment in the power consumption of structures (such as IMRG) designed for adjustable power is higher than power increment of the structures (such as CDS) that are designed for non-adjustable power.

Figure 13 shows that, when the graph is dense, i.e., the square containing all the nodes is small in this case, CDS even performs better than MST because in dense graphs the number of non-leaf nodes of CDS is much less than that of MST. We found that when $c = 0.3$, the structure IMRG-CDS performs the best among all structures discussed in this paper. Generally, we found that the relative performance of structure IMRG-CDS improves when the overhead cost c increases. Notice that, structure IMRG-

CDS already performs well when $c = 0$; see illustration of Figure 12.

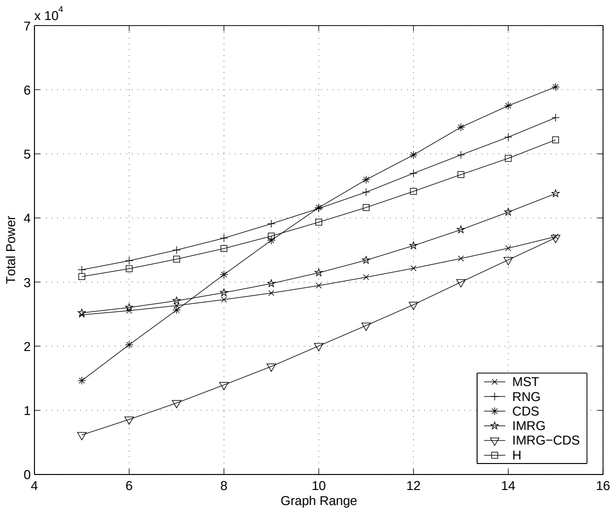


Fig. 13. Power model $c + \|uv\|^{\beta}$ when $c = 0.3$.

V. Virtual Backbone Construction in MANETs using Adjustable Transmission Ranges

ADD this to the organization part

Wu proposed distributed solution based on reducing the network density through a special method that merges two mechanisms: clustering and adjustable transmission range. The basic idea is to first reduce the network density through clustering using a short transmission range. Then neighboring clusterheads (i.e., clusterheads that are 2 or 3 hops away) are connected using a long (and normal) transmission range. In this way, neighboring clusterheads are connected without using any gateway selection process. Connected clusterheads form a CDS. Depending on the selection of the short and long transmission ranges, two versions of the distributed solution are given. A pruning process can be applied on the connected clusterhead set to further reduce the size of the CDS.

With the use of adjustable transmission range, another objective is also achieved: energy-efficient design, which is important in MANETs, because each node is operated on battery with limited capacity. In fact, the proposed energy-efficient design also achieves several other goals as by-products: reducing computation complexity of the broadcast algorithm, maximizing traffic capacity of the network, reducing power consumption of the broadcast process, prolonging life span of each individual node, and reducing contention at the MAC layer.

A. Backbone Formation in Dense Networks

In this section, a density reduction approach which can be integrated to any local approach for CDS construction is proposed. In the proposed methods, the network density is first reduced through clustering using a short transmission range. Then neighboring clusterheads are connected using a long (and normal) transmission range. In this way, neighboring clusterheads are connected without using gateways and form a CDS. This CDS consisting of all clusterheads can be further reduced by applying *marking process* introduced by Jie Wu and Hailan Li [52]. Depending on the selection of the short and long transmission ranges, two approaches can be used to construct a backbone. The first approach adopts a *2-level hierarchy*: In the lower level, the entire network is covered by the set of clusterheads under the short transmission range. In the upper level, all clusterheads are covered by the set of marked clusterheads under the long transmission range. The second approach constructs a *flat backbone*, where the entire network is directly covered by the set of marked clusterheads with the long transmission range. For each approach, an efficient broadcast scheme as an application will be shown.

A.1 2-level clustering approach

We first use different transmission ranges at different stages of protocol handshake, and then apply the long (and normal) transmission range in broadcasting among clusterheads and the short transmission range in broadcasting within each cluster with an unmarked clusterhead. This approach is similar to the clustering approach that forms a CDS in a dense graph. However, unlike the regular clustering approach where a selection process is needed to select gateway nodes to connect clusterheads, we use a reduced transmission range for clustering. The virtual backbone formation procedure is as follows:

Marking process on clusterheads:

1. Each node uses a transmission range of $\frac{1}{3}r$ for cluster formation.
2. Each clusterhead uses a transmission range of r for marking process [52].

In the above process, the backbone is constructed based on clusterheads using a transmission range of $\frac{1}{3}r$. A transmission range of $\frac{1}{3}r$ ensures that all neighboring clusterheads (i.e., clusterheads within 3 hops) are directly connected under a transmission range of r. More formally, we use $G = (V; r)$ to represent a unit disk graph with node set V and r represents a uniform transmission range. Two nodes are connected if their Euclidean distance is no more than r. G can be simplified to $G(r)$ to represent a unit disk graph with a uniform transmission range of r. It is assumed that the graph is sufficiently dense such that $G(\frac{r}{k})$ is still a connected graph for a small k such as $k = 3$ or 4.

Theorem 4: The clusterhead set V' , derived from $G(\frac{r}{3})$ via clustering, is a CDS of $G(r)$.

Let $G'(r)$ be the subgraph of $G(r)$ derived from V'. Since marking process preserve a CDS, we have

Corollary 5: V'' derived from the marking process is a CDS of $G'(r)$.

The broadcast process is as follows:

Broadcast process:

1. If the source is a non-clusterhead, it transmits the message with a transmission range of $\frac{r}{3}$ to the source clusterhead.

2. The source clusterhead transmits the message with a transmission range of r.
3. At each intermediate node, if the node is a marked clusterhead, it forwards the message with a transmission range of r and if it is an unmarked clusterhead, it forwards the message with a transmission range of $\frac{r}{3}$; otherwise, it does nothing.

Theorem 6: The broadcast process ensures full coverage.

When the notion of clusterhead coverage is extended to cover clusterheads and all their members, each unmarked clusterhead is still required to forward the message with a transmission range of $\frac{r}{3}$ to ensure coverage within its cluster, because when marking process is used, the coverage is only extended to all clusterheads, not to all their members which are within $\frac{r}{3}$.

put figure here

A.2 1-level flat approach

In the 2-level clustering approach, the broadcast process involves both *inter-clustering* and *intra-clustering* broadcast using different transmission ranges. In the 1-level flat approach, the notion of clustering is removed by using a uniform transmission range. Still, different transmission ranges are used at different stages of protocol handshake. The modified cluster formation procedure is as follows:

Marking process on clusterheads

1. Each node uses a transmission range of $\frac{r}{4}$ for cluster formation.
2. Each clusterhead uses a transmission range of $\frac{3.r}{4}$ for marking process.

Theorem 7: The clusterhead set V' , derived from $G(\frac{r}{4})$ via clustering, is a CDS of $G(\frac{3.r}{4})$.

Corollary 8: V'' derived from marking process is a CDS of $G'(\frac{r}{4})$.

As a result of the above process, marked clusterheads form a CDS among all nodes in the network. The broadcast process is as follows:

Broadcast process:

Each node uses a transmission range of r for a blind flooding on marked clusterheads.

Theorem 9: The broadcast process ensures full coverage.

The density reduction approach can also be used in other local algorithms for CDS construction. For example, in multipoint relay (MPR) [kkk], each node collects 2-hop neighbor information and then selects a subset of 1-hop neighbors to cover its 2-hop neighbor set. The selected nodes form a CDS.We can use a small transmission range to select clusterheads/cores in a dense network and then use large transmission range(s) for 1-hop and 2-hop neighbor set collection and transmission. The difference is that MPR, instead of marking process, is used in the second stage to further reduce the size of the CDS.

A.3 Performance analysis

The quality of a backbone is measured by the approximation ratio, i.e., an upper bound of the ratio between the size of the backbone to the size of the minimal CDS. This subsection shows that both approaches have $O(1)$ approximation ratio, and $O(\Delta)$ computation complexity and $O(1)$ message complexity at each node, where Δ is the maximum node degree in the network. We also analyze time steps (or rounds of control message exchange) used in the CDS formation. Although the proposed approaches need $O(n)$ rounds in the worst case, where n is the number of nodes in the network, we show that they complete in $O(logn')$ rounds in most cases, where n' is the number of clusterheads and is usually proportional to the area of the 2-D space occupied by a MANET, and reversely proportional to the transmission range. Note that both proposed approaches consist of two stages: (1) cluster formation and (2) pruning via marking process. The $O(1)$ approximation ratio is guaranteed by stage 1 and preserved in stage 2. That is, an upper bound exists on the number of clusterheads in a finite area. Assume transmission range r_1 is used in stage 1 and r_2 in stage 2. We call node v a neighboring clusterhead of node u, if v is a clusterhead in stage 1 and within range r_2 of u. The following lemma shows that the number of neighboring clusterheads is bounded by a constant. A similar lemma has been proved in [kkk].

Lemma 10: Each node has at most $\left(\frac{r_1+2r_2}{r1}\right)^2$ neighboring clusterheads.

Theorem 11: The 2-level clustering approach has an approximation ratio of 49. The 1-level flat approach has an approximation ratio of 81.

Note, however, that the importance of the approximation ratio, which gives a bound on the worst case performance of a CDS algorithm, should not be overstated. The average performance under random networks, which is a more important metric, can only be obtained via probabilistic analysis or simulation study.

Theorem 12: Both proposed approaches have $O(\Delta)$ computation complexity and $O(1)$ message complexity at each node, where Δ is the maximal node degree under the transmission range used in the cluster formation stage.

Note that we assume a constant length for node id in Theorem 12. When n is extremely large, it takes $O(logn)$ bits to represent a unique node id and $O(logn)$ time to process each message. In this case, the proposed approaches have $O(\Delta logn)$ computation complexity and $O(logn)$ message complexity at each node.

Another measure of the time is the number of rounds of message exchanges. In a MANET with dynamic topology changes, a CDS is formed and maintained via periodic exchange of control messages among neighbors. Due to the interdependence among control messages from different nodes, a CDS formation process usually requires several rounds. For example, marking process completes in two rounds. In the first round, each node advertises its id. In the second round, each node advertises its 1-hop neighbor set built in the last round. Then the status of each node can be determined based on its neighbors neighbor sets.

Unfortunately, cluster formation may not complete in constant rounds. Assume clusterheads are elected with minimal node id. In the best case, stage 1 completes in 3 rounds: After every node advertises its id, all clusterheads are elected in the second round, and all non-clusterheads announce their status in the third round. In the worst case, stage 1 may take $O(n)$ rounds. As shown in Figure **include figure**, when all nodes form a sequence with decreasing node ids (i.e., $v_1 > u_1 > v_2 > u_2 .... > v_l$), the cluster formation process requires $n + 1$ rounds to complete. Node v_1 cannot become a clusterhead until u_1 becomes a non-clusterhead, while before u_1 becomes a non-clusterhead, it must wait for v_2 to become a clusterhead, and so on. Fortunately, the following theorem shows that the situation is much better in the average case.

Theorem 13: Let K be the number of rounds used in a cluster formation process and n' the number of clusterheads elected, The expectation of K, $E[K] = O(log(n'))$.

B. *Extension*

In this section, we first extend the backbone formation scheme in the previous section to a general framework, where other DS and CDS formation algorithms can be used to substitute cluster formation and the marking process. This framework is further generalized to support multiple layers of density reduction in very dense networks.

B.1 A general framework

Both 2-level clustering and 1-level flat approaches can be generalized into the following 2-stage process:

2-stage backbone formation
1. Each node uses a transmission range of r_1 to form a DS, V' , of $G(r_1)$.
2. Each node in V' uses a transmission range of $r_2 = 3r_1$ to form a CDS, V'', of $G'(r_2)$.

Here $G'(r_2)$ is the subgraph of $G(r_2)$ induced by V' . In stage 1, any algorithm that yields a dominating set can be used.

Theorem 14: If $G(r_1)$ is connected, V'' is a CDS of both $G'(r_2)$ and $G(r_2 + r_1)$.

The following broadcast schemes show two different ways of using the backbone formed by the above process.

2-level broadcast process:
1. The source node and all nodes in V'' transmit the message with a transmission range of r_2.
2. All other nodes in V' transmit the message with a transmission range of r_1.

1-level broadcast process:
The source node and all nodes in V'' transmit the message with a transmission range of $r_2 + r_1 = 4r_1$.

The following algorithm shows the correctness of above approaches.

Theorem 15: If $G(r_1)$ is connected, both 2-level and 1-level broadcast processes ensure full coverage.

B.2 Recursive density reduction

When the network is very dense, each node may still have a large node degree with the transmission range r_1, which causes high contention and computation cost in stage 1. In this case, the above 2-stage process can be further generalized into the following k-stage process, where the node degree is reduced in stage 1 using a smaller r_1, and bounded by a constant in all subsequent stages. The resultant backbone can be used in both hierarchical routing (as demonstrated by the k-level broadcast process) and flat routing (as demonstrated by the 1-level broadcast process).

---

k-stage backbone formation:
- 1.Each node uses a transmission range of r_1 to form a DS, V_1, of $G(r_1)$.
- 2.Each node uses a transmission range of $r_2 = 3r_1$ to form a DS, V_2, of $G_1(r_2)$.
- ...
- k.Each node in V_{k-1} uses a transmission range of $r_k = 3r_{k-1}$ to form a CDS, V_k, of $G_{k-1}(r_k)$.

Theorem 16: If $G(r_1)$ is connected, V_k is a CDS of both $G_{k-1}(r_k)$ and $G(r_k + r_{k-1} + ... + r_1)$.

---

k-level broadcast process:
- 1.The source node and all nodes in V_k transmit the message with a transmission range of r_k.
- 2.All other nodes in V_{k-1} transmit the message with a transmission range of r_{k-1}
- ...
- k. All other nodes in V_1 transmit the message with a transmission range of r_1.

1-level broadcast process:
The source node and all nodes in V_k transmit the message with a transmission range of $r_k + r_{k-1} + ... + r_1 = \frac{3^k-1}{2} r_1$.

Theorem 17: If $G(r_1)$ is connected, both k-level and 1-level broadcast processes ensure full coverage.

VI. Conclusion

In this paper, we reviewed several methods for efficient broadcasting for wireless ad hoc networks. Although the structures IMRG and LMST$_k$ ($k \geq 2$) have total edge length within a constant factor of the MST, the broadcasting based on these locally constructed structures could still consume energy arbitrarily larger than the optimum in the worst case, when we assume that the power needed to support a link uv is $\|uv\|^\beta$. It has been proved that the broadcasting based on MST consumes energy within a constant factor of the optimum, but MST cannot be constructed locally. It is already known that no structure can be constructed locally and the broadcasting based on it consumes energy within a constant factor of the optimum when the power needed to support a link uv is $\|uv\|^\beta$. Although it is known that MST and BIP consumes power within a constant factor of the minimum, when the communication overhead for maintenance is added, localized solution becomes superior.

In this paper, we proposed a new structure, called IMRG-CDS, that can be constructed efficiently in a localized manner using only $O(n)$ communications under the local broadcast communication model, i.e., assuming the message sent by a node can be received by all nodes within its transmission range. We conducted extensive simulations to study the performance of various structures for broadcasting: structures constructed in a centralized manner, such as *MST* and *BIP*, and some structures constructed in a localized manner, such as *RNG, IMRG, CDS*, *LMST*, and our new structure IMRG-CDS. In our simulations, we compared their performances in both adjustable and non-adjustable power consumption model. We also compared the cost of constructing these structures, i.e. number of messages used to build the structure, for localized methods. We found that our new structure IMRG-CDS performs well in both power models. When the power needed to support a link uv is $c+\|uv\|^{\beta}$, we found that the structure IMRG-CDS performs better and better when c increases.

References

[1] Deborah Estrin, Ramesh Govindan, John S. Heidemann, and Satish Kumar, "Next century challenges: Scalable coordination in sensor networks," in *Mobile Computing and Networking (Mobicom)*, 1999, pp. 263–270.

[2] J. M. Kahn, R. H. Katz, and K. S. J. Pister, "Next century challenges: Mobile networking for smart dust," in *International Conference on Mobile Computing and Networking (MOBICOM)*, 1999, pp. 271–278.

[3] Seapahn Megerian and Miodrag Potkonjak, "Wireless sensor networks," in *Book Chapter in Wiley Encyclopedia of Telecommunications, Editor: John G. Proakis*, 2002.

[4] G.J. Pottie and W.J. Kaiser, "Wireless integrated network sensors," in *Communications of the ACM, vol. 43, no. 5*, 2000, pp. 551–558.

[5] Chalermek Intanagonwiwat, Ramesh Govindan, and Deborah Estrin, "Directed diffusion: a scalable and robust communication paradigm for sensor networks," in *Mobile Computing and Networking (Mobicom)*, 2000, pp. 56–67.

[6] Seapahn Meguerdichian, Farinaz Koushanfar, Miodrag Potkonjak, and Mani Srivastava, "Coverage problems in wireless ad-hoc sensor network," in *IEEE INFOCOM'01*, 2001, pp. 1380–1387.

[7] S. Capkun, M. Hamdi, and J.P. Hubaux, "Gps-free positioning in mobile ad-hoc networks," in *Proc. Hawaii Int. Conf. on System Sciences*, 2001.

[8] C. Chiang and M. Gerla, "demand multicast in mobile wireless networks," 1998.

[9] Jorjeta G. Jetcheva, Yih-Chun Hu, David A. Maltz, and David B. Johnson, "A simple protocol for multicast and broadcast in mobile ad hoc networks," 2001.

[10] M. Liu, R. Talpade, and A. McAuley, "Amroute: Adhoc multicast routing protocol," 1999.

[11] Charles E. Perkins, Elizabeth M. Belding-Royer, and Samir Das, "Ad hoc on demand distance vector (aodv) routing," 2002.

[12] Y.-C. Tseng, Y.-S. Chen S.-Y. Ni, and J.-P. Sheu, "The broadcast storm problem in a mobile ad hoc network," in *Wireless Networks, vol. 8 (Short version in MOBICOM 99)*.

[13] B. Williams and T. Camp, "Comparison of broadcasting techniques for mobile ad hoc networks," in *Proceedings of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC)*, 2002, pp. 194–205.

[14] C. W. Wu, Y. C. Tay, , and C.-K Toh, "Ad hoc multicast routing protocol utilizing increasing id-numbers (amris) functional specification," 1998.

[15] J. Wu and W. Lou, "Forward-node-set-based broadcast in clustered mobile ad hoc networks," in *Wireless Communications and Mobile Computing*, 2003, pp. 155–173.

[16] Khaled Alzoubi, Peng-Jun Wan, and Ophir Frieder, "Message-optimal connected-dominating-set construction for routing in mobile ad hoc networks," in *3rd ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc'02)*, 2002.

[17] J. Wieselthier, G. Nguyen, and A. Ephremides, "On the construction of energy-efficient broadcast and multicast trees in wireless networks," in *Proc. IEEE INFOCOM 2000*, 2000, pp. 586–594.

[18] T. J. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to Algorithms*, MIT Press and McGraw-Hill, 1990.

[19] A. Clementi, P. Crescenzi, P. Penna, G. Rossi, and P. Vocca, "On the complexity of computing minimum energy consumption broadcast subgraphs," in *18th Annual Symposium on Theoretical Aspects of Computer Science, LNCS 2010*, 2001, pp. 121–131.

[20] Peng-Jun Wan, G. Calinescu, Xiang-Yang Li, and Ophir Frieder, "Minimum-energy broadcast routing in static ad hoc wireless networks," *ACM Wireless Networks*, 2002, To appear. Preliminary version appeared in IEEE INFOCOM 2000.

[21] P. Berman, M. Furer, and A. Zelikovsky, "Applications of matroid parity problem to approximating steiner trees," Tech. Rep. 980021, Computer Science, UCLA, 1998.

[22] G. Robins and A. Zelikovsky, "Improved steiner tree approximation in graphs," in *Proceedings of ACM/SIAM Symposium on Discrete Algorithms*, 2000, pp. 770–779.

[23] Harry B. Hunt III, Madhav V. Marathe, Venkatesh Radhakrishnan, S. S. Ravi, Daniel J. Rosenkrantz, and RichardE. Stearns, "NC-approximation schemes for NP- and PSPACE -hard problems for geometric graphs," *J. Algorithms*, vol. 26, no. 2, pp. 238–274, 1999.

[24] Khaled M. Alzoubi, *Virtual Backbone in Wireless Ad Hoc Networks*, Ph.D. thesis, Illinois Institute of Technology, 2002.

[25] Xiuzhen Cheng, "A polynomial time approximation algorithm for connected dominating set problem in wireless ad hoc networks," 2002.

[26] Khaled M. Alzoubi, Peng-Jun Wan, and Ophir Frieder, "New distributed algorithm for connected dominating set in wireless ad hoc networks," in *HICSS, Hawaii*, 2002.

[27] Chunhung Richard Lin and Mario Gerla, "Adaptive clustering for mobile wireless networks," *IEEE Journal of Selected Areas in Communications*, vol. 15, no. 7, pp. 1265–1275, 1997.

[28] I. Chlamtac and A. Farago, "A new approach to design and analysis of peer to peer mobile networks," *Wireless Networks*, vol. 5, pp. 149–156, 1999.

[29] Alan D. Amis, Ravi Prakash, Dung Huynh, and Thai Vuong, "Max-min d-cluster formation in wireless ad hoc networks," in *Proc. of the Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies INFOCOM*, 2000, vol. 1, pp. 32–41.

[30] Ji-Cherng Lin, Shi-Nine Yang, and Maw-Sheng Chern, "An efficient distributed algorithm for minimal connected dominating set problem," in *Proc. of the Tenth Annual International Phoenix Conference on Computers and Communications 1991*, 1991, pp. 204–210.

[31] Alan D. Amis and Ravi Prakash, "Load-balancing clusters in wireless ad hoc networks," in *Proc. of the 3rd IEEE Symposium on Application-Specific Systems and Software Engineering Technology*, 2000.

[32] Ashwinder Ahluwalia, Eytan Modiano, and Li Shu, "On the complexity and distributed construction of energy efficient broadcast trees in static ad hoc wireless networks," in *Proceedings of 36th Annual Conference on Information Sciences and Systems (CISS)*, 2002.

[33] Mario Cagalj, Jean-Pierre Hubaux, and Christian Enz, "Minimum-energy broadcast in all-wireless networks: Np-completeness and distribution issues," in *Proceedings of the ACM MOBICOM 02*, 2002.

[34] J.E. Wieselthier, G.D. Nguyen, and A. Ephremides, "The energy efficiency of distributed algorithms for broadcasting in ad hoc networks," in *Proceedings of IEEE 5th International Symposium on Wireless Personal Multimedia Communication (WPMC)*, 2002, pp. 499–503.

[35] Ning Li and Jennifer C. Hou, "Blmst: a scalable, power efficient broadcast algorithm for wireless sensor networks (submitted for publication, a version as uiuc computer science department technical report.)," in *2003*.

[36] G. Toussaint, "The relative neighborhood graph of finite planar set," in *Pattern Recognition vol. 12, no. 4*, 1980, pp. 261–268.

[37] J. Cartigny, D. Simplot, and I. Stojmenvic, "Localized

minimum-energy broadcasting in ad-hoc networks," in *Proc. of IEEE INFOCOM 2003*, 2003.

[38] Ivan Stojmenovic, Mahtab Seddigh, and Jovisa Zunic, "Dominating sets and neighbor elimination based broadcasting algorithms in wireless networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 13, no. 1, pp. 14–25, 2002.

[39] J. Carle, D. Simplot, and I. Stojmenovic, "Area coverage based beaconless broadcasting in adhoc networks (in preparation.)," .

[40] Ning Li, Jennifer C. Hou, and Lui Sha, "Design and analysis of a mst-based topology control algorithm," in *Proc. of IEEE INFOCOM 2003*, 2003.

[41] Xiang-Yang Li, "Efficient code assignment for wireless networks," 2002, Submitted for publication.

[42] Xiang-Yang Li, Yu Wang, Peng-Jun Wan, and Ophir Frieder, "Localized low weight graph and its applications in wireless ad hoc networks," 2003.

[43] Gruia Călinescu, Ion Măndoiu, Peng-Jun Wan, and Alexander Zelikovsky, "Selecting forwarding neighbors in wireless ad hoc networks," in *ACM DialM*, 2001.

[44] Gruia Călinescu, "Computing 2-hop neighborhoods in ad hoc wireless networks," 2002, Submitted for publication.

[45] Peng-Jun Wan, Khaled M. Alzoubi, and Ophir Frieder, "Distributed construction of connected dominating set in wireless ad hoc networks," in *INFOCOM*, 2002.

[46] Yu Wang and Xiang-Yang Li, "Geometric spanners for wireless ad hoc networks," in *Proc. of 22nd IEEE International Conference on Distributed Computing Systems (ICDCS)*, 2002, To appear.

[47] Xiang-Yang Li, G. Calinescu, and Peng-Jun Wan, "Distributed construction of planar spanner and routing for ad hoc wireless networks," in *21st Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, 2002, vol. 3.

[48] Ivan Stojmenovic, "Degree limited rng," in *Workshop on Wireless Networks*.

[49] Xiang-Yang Li, Ivan Stojmenovic, and Yu Wang, "Partial delaunay triangulation and degree limited localized bluetooth multihop scatternet formation," in *IEEE Transactions on Parallel and Distributed Systems*, 2003.

[50] Gautam Das, Giri Narasimhan, and Jeffrey Salowe, "A new way to weigh malnourished euclidean graphs," in *ACM Symposium of Discrete Algorithms*, 1995, pp. 215–222.

[51] Mahtab Seddigh, J. Solano Gonzalez, and I. Stojmenovic, "Rng and internal node based broadcasting algorithms for wireless one-to-one networks," *ACM Mobile Computing and Communications Review*, vol. 5, no. 2, pp. 37–44, 2002.

[52] Jie Wu and Hailan Li, "A dominating-set-based routing scheme in ad hoc wireless networks," *Telecommunication Systems*, vol. 18, no. 1–3, pp. 13–36, 2001.