A Distributed Formation of a Virtual Backbone in MANETs using Adjustable Transmission Ranges [*]

Jie Wu and Fei Dai
Department of Computer Science and Engineering
Florida Atlantic University
Boca Raton, FL 33431

Abstract

Recently, the use of a virtual backbone (connected dominating set) in various applications in mobile ad hoc networks (MANETs) has become popular. These applications include topology management, point and area coverage, and routing protocol design. In a mobile environment such as a MANET, one challenging issue in constructing a virtual backbone is to accomplish a distributed and localized solution that aims at balancing several contradicting objectives: small approximation ratio, fast convergence, and low computation cost. Many existing distributed and localized algorithms select a virtual backbone without resorting to global or geographical information. However, these algorithms incur a high computation cost in a dense network. In this paper, we propose a distributed solution based on reducing the density of the network using two mechanisms: clustering and adjustable transmission range. By using adjustable transmission range, we also achieve another objective, energy-efficient design, as a by-product. As an application, we show an efficient broadcasting where nodes (and only nodes) in a virtual backbone are used to forward the broadcast message. The virtual backbone is constructed using Wu and Li's marking process [1] and the proposed density reduction process. The application of the density reduction process to other localized algorithms is also discussed. The efficiency of our approach is confirmed through both analytic and simulation study.

1. Introduction

Although a mobile ad hoc network (or simply MANET) has no physical backbone infrastructure, a *virtual backbone* can be formed by nodes in a *connected dominating set* (CDS) of the unit-disk graph of a given MANET. Recently, the use of a virtual backbone in various applications in MANETs has become popular. These applications include topology management in MANETs, point and area coverage in sensor networks, and routing protocol design. A dominating set (DS) is a subset of nodes in the network where every node is either in the subset or a neighbor of a node in the subset. In a unit-disk graph, node connections are determined by their geographical distances. It has been proved that finding the minimum CDS in a unit-disk graph is NP-complete.

A common source of overhead in a MANET comes from blind flooding/broadcasting, where a broadcast message is forwarded by every node exactly once. Broadcasting is used in the route discovery process in several reactive routing protocols. Due to the broadcast nature of wireless communication (i.e., when a source sends a message, all its neighbors will hear it), blind flooding/broadcasting may generate excessive redundant transmission. Redundant transmission may cause a serious problem, referred to as the broadcast storm problem [2], in which redundant messages cause communication contention and collision. In Figure 1 (a), when each node forwards the message once, node w will receive the same message six times. To reduce redundant transmission, nodes (and only nodes) in the virtual backbone forward the message once when they receive the message for the first time.

In a mobile environment such as a MANET, one challenging issue in constructing a virtual backbone is to accomplish a distributed and localized solution that aims at balancing several contradicting objectives: small approximation ratio, fast convergence, and low computation cost. Many existing distributed and localized algorithms can select a virtual backbone without resorting to global or geographical information. However, these algorithms incur a high computation cost in a dense network. For example, in Wu and Li's marking process [1], each node is marked (i.e., in CDS) if it has two unconnected neighbors. The marking process is rather effective in reducing the size of the CDS. In addition, it supports localized maintenance in a mobile environment. However, the process incurs a high computation cost in a dense network since each node needs to check all pairs of its neighbors.

In this paper, we propose a distributed solution based on reducing the network density using *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

[*] This work was supported in part by NSF grants CCR 0329741, ANI 0073736, and EIA 0130806. Contacting address: jie@cse.fau.edu

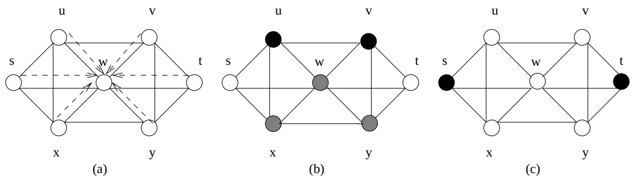


Figure 1. (a) Broadcast storm problem. (b) Marked nodes: gray (marked by the marking process and then unmarked by Rules 1 and 2) and black (marked by the marking process). (c) Clustering approach: black nodes (clusterheads) and white nodes (non-clusterheads).

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.

As an application, we show an efficient broadcasting where the virtual backbone is constructed using the clustering approach, followed by Wu and Li's marking process as the pruning process on the clusterhead set. Note that the density reduction approach can be used in other localized solutions such as multipoint relay (MPR) [3]. With the use of adjustable transmission range, we also achieve another objective, 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.

2. Related work

Wu and Lou [4] gave a comprehensive classification of CDS construction algorithms in MANETs: *global*, *quasi-global*, *quasi-local*, and *local*. Global solutions are based on global state information while local solutions use only local state information. In *localized* solutions [5], nodes interact with others in the neighborhood. Each node performs exceedingly simple tasks of maintaining and propagating information *markers*. If the propagation is bounded by a small constant, the corresponding solution is local; if the propagation in general is bounded by a small constant, but with occasional long sequential propagation, the corresponding protocol is called quasi-local. Quasi-global does not use global information, but relies on a global infrastructure such as a tree that must be constructed through global sequential propagation. The global solutions include Guha and Khuller's approximation algorithm [6] and have been used in protocol design by Das et al [7]. The quasi-global solution includes Alzoubi et al's SPT approach [8]. The clustering approach (to be discussed in detail in the next section) falls into the quasi-local category. Clusters are formed by first electing a clusterhead whose neighbors then join in the cluster as non-clusterhead members. Local solutions include Wu and Li's marking process (MP) [1], several variations [9, 10] of MP, Qayyum, Viennot, and Laouiti's multipoint relay (MPR) [3].

The formation of a CDS is sometimes tied with a broadcast process. Wu and Dai [11] classified broadcast algorithms that form a CDS using local solutions as *self-pruning* and *neighbor-designating*. In self-pruning methods [1, 9, 10, 12], each node makes its local decision on its status: forwarding (i.e., within the CDS) or non-forwarding (i.e., outside the CDS). In neighbor-designating methods [3], the status of each node is determined by its neighbors. Local methods also have the following two orthogonal classifications based on the way the CDS is constructed: static (before the broadcast process) vs. dynamic (during the broadcast process), and source-independent (independent of the location of the source) vs. source-dependent (dependent on the location of the source). In general, dynamic is better than static in terms of generating a small CDS. Similarly, source-dependent edges out source-independent. However, neither dynamic nor source-dependent methods produce a general purpose CDS – a new CDS is constructed for each source and/or broadcast process.

Energy-efficient broadcast has also been widely studied and is sometimes associated with topology control protocols. Several protocols have been proposed to manage energy consumption by adjusting transmission ranges. For a comprehensive survey on various aspects of broadcasting in ad hoc networks, refer to [13]. In this paper, we use the static and source-independent approach for CDS construction since it is more generic. The resultant CDS is suitable for all situations. It is further assumed that no location information is provided.

3. Preliminaries

3.1. Marking process

Wu and Li [1] proposed a self-pruning process, called **marking process**, to construct a CDS: *Each node is marked if it has two unconnected neighbors; otherwise, it is unmarked.*

The marked nodes form a CDS, which can be further reduced by applying two rules for pruning (i.e., changing a marked node back to an unmarked node). A set U is said to be *covered* by V (and V is called a *coverage set* of U) if every node in U is either in V or a neighbor of a node in V. According to pruning Rule 1, a marked node can unmark itself if its neighbor set is covered by another marked node with a higher priority; that is, if all its neighbors are connected with each other via another dominating node with a higher priority, it can be safely removed from the CDS. The node priority can be defined based on node degree (which is dynamic) and/or node id (which is static). According to pruning Rule 2, a marked node can unmark itself if its neighborhood is covered by two other directly connected marked nodes with higher priorities. The combination of Rules 1 and 2 is fairly efficient in reducing the size of a CDS. Dai and Wu [10] further extended the pruning rule to **pruning rule** k: *A marked node can unmark itself if its neighbor set is covered by a set of connected nodes with higher priorities.*

When the coverage set is restricted to a subset of the neighbor set, the corresponding rule is called a *restricted rule*. It

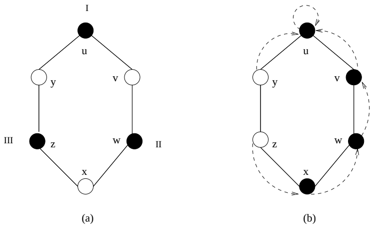


Figure 2. The clustering approach with black nodes as clusterheads in (a) and cores in (b).

has been shown in [10] that a restricted rule is almost as efficient as the original non-restricted rule in reducing the size of the CDS. It has also been shown that the restricted pruning Rule k can be implemented with the same cost as the restricted Rule 1 ($O(\Delta^2)$), but with less cost than the restricted Rule 2 ($O(\Delta^3)$), where Δ is the maximal node degree in the network. In the subsequent discussion, we use Rule k to refer to the restricted pruning Rule k. Note that both marking process (MP) and the restricted pruning Rule k require only 2-hop information at each node. However, to apply MP and Rule k, each node needs to check $O(\Delta^2)$ pairs of neighbors, making it costly in dense networks.

Figure 1 (b) shows the example of MP and Rules 1 and 2 with node id as the priority; that is, the lower the id of a node, the higher the priority of the node (e.g., u has a higher priority than w). Nodes u, v, w, x, and y are marked after applying MP. Nodes x and y are unmarked by Rule 1, since their neighbor sets are covered by w. Then w is unmarked by Rule 2, since its neighbor set is jointly covered by u and v which are directly connected.

3.2. Clustering approach

The clustering approach is commonly used to offer scalability and is efficient in a dense network. Basically, the network is partitioned into a set of clusters, with one clusterhead in each cluster. Clusterheads form a DS, but no two clusterheads are connected. Each clusterhead directly connects to all its members (also called non-clusterheads). The classical clustering **cluster formation** works as follows: *(1) A node v is a* clusterhead *if it has the highest priority (smallest node id or maximum node degree) in its 1-hop neighborhood including v. (2) A clusterhead and its neighbors form a cluster and these nodes are* covered*. (3) Repeat (1) and (2) on all uncovered nodes (if any).*

Figure 1 (c) shows an example of the clustering process. Both s and t are clusterheads (black nodes) since they are local minima. u and x belong to cluster s while v and y belong to cluster t. Node w can belong to either s or t. If the node id of w is changed to m in Figure 1 (c), node m is the only clusterhead. When a node has multiple adjacent clusterheads, it belongs to one of them. Note that the cluster formation may need several rounds to complete depending on the network topology and the priority distribution.

Once the cluster formation process completes, non-clusterheads are designated as *gateways* to connect clusterheads. In [14], a global infrastructure (such as a tree) is used to select a small set of gateways with a constant approximation ratio of 12. In this scheme, a clusterhead is first determined as the root through a distributed election, and then the root initiates a global flooding to build a tree that connects all clusterheads. In most localized approaches [4, 15], clusterheads are connected via a mesh, rather than a tree structure. Using the greedy approach, each clusterhead designates a set of gateways to connect all neighboring clusterheads (i.e., clusterheads 2 or 3 hops away). Optimization is possible for each clusterhead to connect clusterheads that have members 1 or 2 hops away while ensuring connectivity among clusterheads [4]. The mesh scheme also has a constant approximation ratio, because in unit disk graphs, each clusterhead has only a finite number of neighboring clusterheads and, therefore, designates a finite number of gateways.

In the core-based approach (used in CEDAR [16]), clusterheads (called core nodes) are permitted to be adjacent, but the core formation can be done in a constant number of rounds without sequential propagation. The original core-based approach is non-deterministic (i.e. time-sensitive depending on when each node participates in the formation process). Here we consider a simplified and deterministic version: *A node v becomes a core node if (1) it has the highest priority among its 1-hop neighbors including v (v is selected by itself as a core node), or (2) it has the highest priority based on a neighbor's 1-hop neighborhood (v is selected by a neighbor as a core node).*

Figure 2 shows the application of both cluster and core formations to the same network. Node degree is used as the priority and node id is used to break a tie in node degree. In this case, the priority in the decreasing order is $u > v > w > x > y > z$. Black nodes are clusterheads/core nodes. In Figure 2 (a), each Roman numeral indicates the round number (assume the formation is synchronous) in which the corresponding node is selected as a clusterhead. Each dashed arrow line in Figure 2 indicates the *selector* of each core node.

Like clusterheads, core nodes can be connected via gateways to connect neighboring core nodes (i.e., core nodes within 3 hops). In CEDAR, each core node designates a *forwarding set* [16] that consists of both core nodes and non-core nodes (gateways) to form a mesh structure. To distinguish these two approaches, the former is called cluster formation, where clusterheads are not adjacent, and the latter is called core formation.

4. Backbone Formation in Dense Networks

Two approaches can be used to construct a backbone. The first approach adopts a 2-level hierarchy: In the lower level, the network is covered by the set of clusterheads under a short transmission range. In the upper level, all clusterheads are covered by the set of *marked clusterheads* under a long transmission range. The second approach constructs a flat backbone, where the network is directly covered by the set of marked clusterheads with the long transmission range.

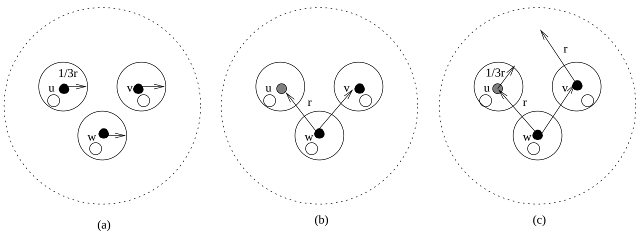


Figure 3. (a) Cluster formation with a range of $1/3r$. (b) Marking process with a range of r. (c) Clusterheads forward the broadcast message with different ranges.

4.1. 2-level clustering approach

We first use different transmission ranges (or simply ranges) at different stages of protocol handshake, and then apply the long (and normal) range in broadcasting among clusterheads and the short 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 range for cluster/core formation.

Marking process on clusterheads

1. Each node uses a range of $1/3r$ for cluster formation.
2. Each clusterhead uses a range of r for MP and Rule k.

In the above process, the backbone is constructed based on clusterheads using a range of $1/3r$. A range of $1/3r$ ensures that all neighboring clusterheads (i.e., clusterheads within 3 hops) are directly connected under a range of r.

More formally, we use $G = (V, P(V), r)$ to represent a unit disk graph with node set V, a mapping $P : V \rightarrow R^2$, where R is the real number set, and $r \in R_+$ represents a uniform range from the positive real number set R_+. P maps each node in V to an (x, y) coordination in 2-D space. 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 range of r. It is assumed that the graph is sufficiently dense such that $G(r/k)$ is still a connected graph for a small k such as $k = 3$ or 4.

Theorem 1: *The clusterhead set $V^{'}$, derived from $G(r/3)$ using either cluster or core formation, is a CDS of $G(r)$.*

Proof: Since $V^{'}$ is a DS of $G(r/3)$, and all links in $G(r/3)$ are preserved in $G(r)$, $V^{'}$ is also a DS of $G(r)$. Then we prove that $V^{'}$ is connected. It is known that in a connected network, all clusterheads (core nodes) can be connected by connecting each clusterhead (core node) to its neighboring clusterheads (core nodes) within 3 hops. It is assumed that $G(r/3)$ is connected. In addition, the range in $G(r)$ is 3 times that of $G(r/3)$. That is, all neighboring clusterheads (core nodes) in $G(r/3)$ are connected in $G(r)$. □

Let $G^{'}(r)$ be the subgraph of $G(r)$ derived from $V^{'}$. Since MP and Rule k preserve a CDS, we have

Corollary 1: *$V^{''}$ derived from the MP and Rule k is a CDS of $G^{'}(r)$.*

Figure 3 (b) shows applying MP and Rule k on clusterheads. The marked clusterheads form a CDS among clusterheads.

Broadcast process

1. If the source is a non-clusterhead, it transmits the message with a range of $1/3r$ to the *source clusterhead*.
2. The source clusterhead transmits the message with a range of r.
3. At each intermediate node, if the node is a marked clusterhead, it forwards the message with a range of r and if it is an unmarked clusterhead, it forwards the message with a range of $1/3r$; otherwise, it does nothing.

Theorem 2: *The broadcast process ensures full coverage.*

Proof: Based on the broadcast process, if the source is not a clusterhead, it will forward the message to its clusterhead. Once the message is received by one clusterhead, it will be forwarded by marked clusterheads in $V^{''}$ to all clusterheads in $V^{'}$ (Corollary 1). Each clusterhead will forward once, using a range of r if it is marked or a range of $1/r$ if it is unmarked. In either case, each clusterhead will cover all members (non-clusterheads) that are within $1/3r$. □

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 range of $1/3r$ to ensure coverage within its cluster, because when MP and Rule k are used, the coverage is only extended to all clusterheads, not to all their members which are within $r/3$. Figure 3 (c) shows the broadcast process in the 2-level clustering approach.

4.2. 1-level flat approach

In the 2-level clustering approach, the broadcast process involves both inter-clustering and intra-clustering broadcast using different ranges. In the 1-level flat approach, the notion of clustering is removed by using a uniform range. Still, different ranges are used at different stages of protocol handshake.

Marking process on clusterheads

1. Each node uses a range of $1/4r$ for cluster formation.
2. Each clusterhead uses a range of $3/4r$ for MP and Rule k.

Theorem 1a: *The clusterhead set $V^{'}$, derived from $G(r/4)$ using either cluster or core formation, is a CDS of $G(3r/4)$.*

Theorem 1a can be proved in the same way as Theorem 1 based on the fact that the range in $G(3r/4)$ is three times that of $G(r/4)$. Let $G^{'}(3r/4)$ be the subgraph of $G(3r/4)$ derived from $V^{'}$, we also have

Corollary 1a: *$V^{''}$ derived from MP and Rule k is a CDS of $G^{'}(3r/4)$.*

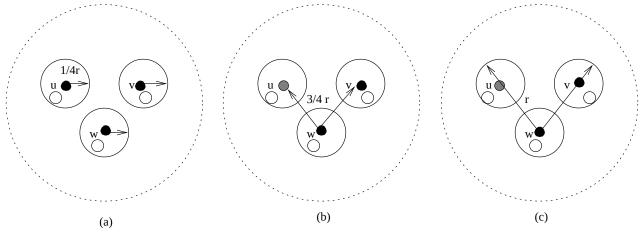


Figure 4. (a) Cluster formation with a range of $1/4r$. (b) Reducing the CDS containing clusterheads with a range of $3/4r$. (c) Only marked clusterheads forward the message with a range of r.

As a result of the above process, marked clusterheads form a CDS among all nodes in the network.

Broadcast process

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

Theorem 2a: *The broadcast process ensures full coverage.*

Proof: Based on the broadcast process, each marked clusterhead in $V^{''}$ forwards the broadcast message. From corollary 1a, each clusterhead u in $V^{'}$ receives the message from at least one neighboring marked cluster v in $G(3r/4)$. Since the distance between u and v is at most $3r/4$ and the distance between u and all its cluster members in $G(r/4)$ is at most $r/4$, the distance from v to each member of u is at most r. That is, all non-clusterheads also receive the broadcast message. □

Figure 4 shows the 1-level clustering approach. Figure 5 illustrates sample backbones constructed via six CDS algorithms. The sample network is in a 100×100 area with 1000 nodes and a normal range of $r = 24$. In Figure 5, MP and Rule k (a) have 72 marked nodes. When the range is r, each node has about 100 neighbors. The large neighbor set causes large control messages and high computation cost. CEDAR (b) has 71 nodes and 60 non-core nodes in forwarding sets. The two cluster-based approaches use gateways to connect clusterheads. The size of the CDS is 33 in the tree scheme (c) and 48 in the mesh scheme (d). The 2-level approach (e) selects 98 clusterheads in the first stage, but only 20 marked clusterheads in the second stage. Note that each clusterhead has only about 10 neighboring clusterheads, which means small control messages and lower computation cost in the second stage. The 1-level flat approach (f) has 156 clusterheads and 43 marked clusterheads. Note that if a clusterhead is within the $3r/4$ range (represented by the dashed circle) of a marked clusterhead in the second stage, then all its members are within the normal range of v (represented by the dotted circle) of this marked clusterhead.

The density reduction approach can also be used in other local algorithms for CDS construction. For example, in multipoint relay (MPR) [3], 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 range to select clusterheads/cores in a dense network and then use large range(s) for 1-hop and 2-hop neighbor set collection and transmission. The difference is that MPR, instead of MP and Rule k, is used in the second stage to further reduce the size of the CDS.

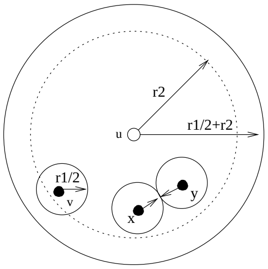


Figure 6. Maximal number of neighboring clusterheads.

5. Performance analysis

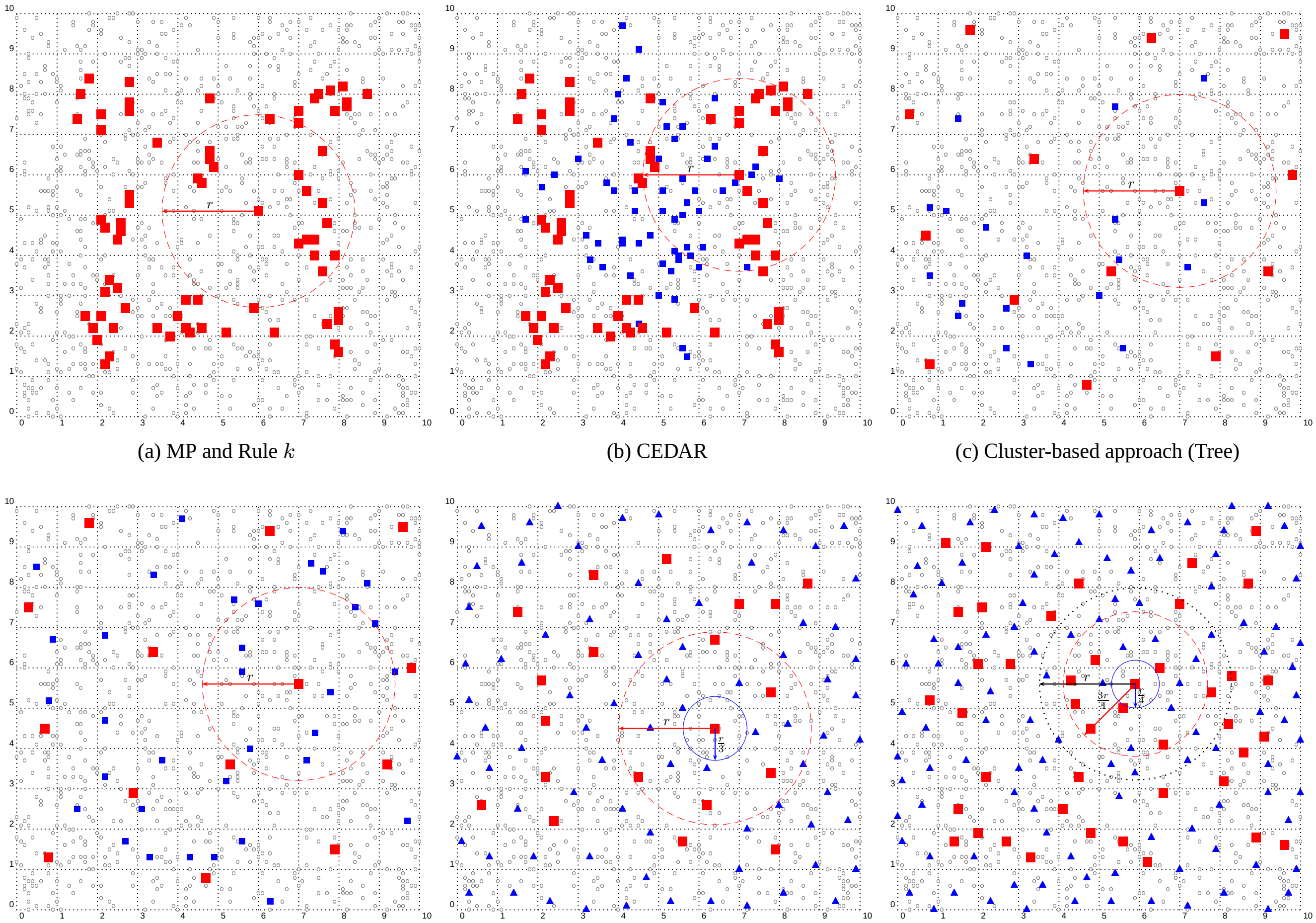
5.1. Complexity analysis

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 $O(\log n')$ 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 MP and Rule k. 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.

Lemma 1: *Each node has at most* $(\frac{r_1 + 2r_2}{r_1})^2$ *neighboring clusterheads.*

Proof: For each clusterhead v, we draw a circle centered at v with radius $r_1/2$ (Figure 6). Because two clusterheads cannot be neighbors, the distance between any two clusterheads is larger than r_1. Those circles with radius $r_1/2$ are non-overlapping. Since the centers of these circles are within range r_2 of u, all these circles are within a large circle centered at u with radius $r_1/2 + r_2$. The total number of neighboring clusterheads of u is no more than the total number of non-overlapping $r_1/2$ circles in the large circle, which is less than $\frac{\pi(r_1/2 + r_2)^2}{\pi(r_1/2)^2} = (\frac{r_1 + 2r_2}{r_1})^2$. □



(a) MP and Rule k

(b) CEDAR

(c) Cluster-based approach (Tree)

(d) Cluster-based approach (Mesh)

(e) 2-level clustering approach

(f) 1-level flat approach

Figure 5. Sample backbones constructed by six CDS algorithms. Squares: nodes in the CDS, small squares: gateways in CEDAR and cluster-based approaches, small triangles: unmarked clusterheads in proposed approaches.

Theorem 3: *The 2-level clustering approach has an approximation ratio of 49. The 1-level flat approach has an approximation ratio of 81.*

Proof: Suppose V_{opt} is a minimal CDS constructed in an optimal approach. The backbone formed by the 2-level clustering approach consists of both marked and unmarked clusterheads. Note that each clusterhead is covered by at least one node in V_{opt}. That is, each clusterhead v elected with $r_1 = r/3$ must have a neighbor $u \in V_{opt}$ within distance $r_2 = r$. Based on Lemma 1, each node in V_{opt} can cover at most 49 clusterheads. Therefore, the number of clusterheads is at most 49 times $|V_{opt}|$. In the 1-level flat approach, the backbone uses marked clusterheads only. By applying Lemma 1 with $r_1 = r_2/4$, the number of clusterheads is less than $81|V_{opt}|$, as is the number of marked clusterheads. □

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 4: *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.*

Proof: In the cluster formation stage, each node sends two $O(1)$ messages, the first containing its id and the second advertising its decision on becoming a clusterhead or non-clusterhead. Each node receives $O(\Delta)$ messages from its neighbors and takes $O(1)$ time in processing each message. Therefore, stage (1) has $O(\Delta)$ computation complexity and $O(1)$ message complexity.

For the pruning stage, it was proved in [10] that both MP and Rule k have $O(\Delta^2)$ computation complexity and $O(\Delta)$ message complexity at each node. As shown in Lemma 1, stage (2) is applied on a sparse graph where $\Delta = O(1)$. Therefore, stage (2) has $O(1)$ time complexity and $O(1)$ message complexity. Overall, both proposed approaches have $O(\Delta)$ computation complexity and $O(1)$ message complexity at each node. □

Here we assume a constant length for node id in Theorem 4. When n is extremely large, each node id uses $O(\log n)$ bits, and the proposed approaches have $O(\Delta \log n)$ computation complexity and $O(\log n)$ message complexity.

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, MP combined with Rule k 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. Fortunately, the following theorem shows that the situation is much better in the average case.

Theorem 5: *For any small constant ε, there exists a function $f(n') = O(\log n')$ such that $P(k < f(n')) > 1 - \varepsilon$, where n' is the number of clusterheads elected and k is the number of rounds used in cluster formation.*

The proof of Theorem 5 can be found in [17]. It shows that in average cases, stage (1) completes in $O(\log n')$ rounds. Since stage (2) requires only two rounds, both proposed approaches complete in $O(\log n')$ rounds in most situations. Note that the number of clusterheads n' in a given 2-D space with area S is bounded by S/r_1^2, where r_1 is the range used in the first stage. Therefore, in the average case, both proposed approaches complete in $O(\log \frac{S}{r_1^2})$ rounds.

5.2. Simulation

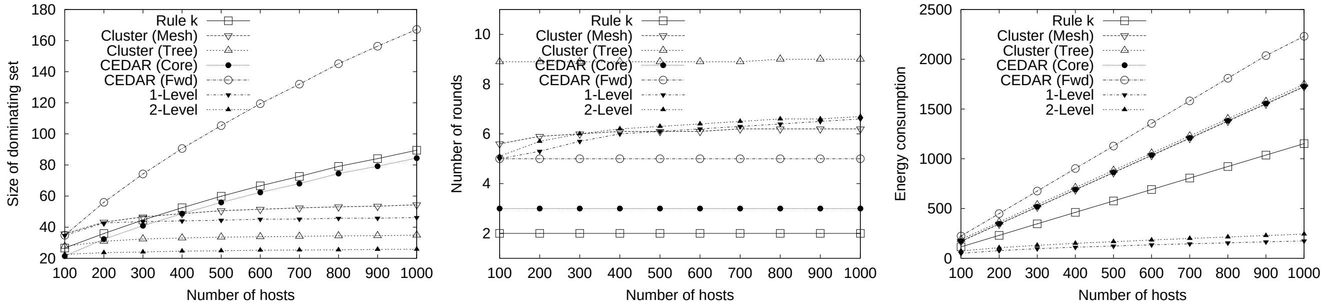
The performance and overhead of both proposed approaches are evaluated via simulations. The 2-level clustering approach (2-Level) and the 1-level flat approach (1-Level) are compared with several existing ones, including the combination of MP and Rule k (Rule k), two cluster-based approaches using a mesh (Mesh) and a tree (Tree) to connect clusterheads, and CEDAR. In the 2-level approach, the resultant CDS is a dominating set of the subnetwork consisting of clusterheads. For CEDAR, two versions are considered: one for the DS consisting of core nodes only (Core) and another for the CDS consisting of both core nodes and non-core nodes in forwarding sets (Fwd). We use node id as priority in cluster formation to reduce the number of packets (2 packets per node for node id while 3 for node degree) and energy consumption. Since Rule k can use node degree as priority with 2 packets per node, node degree is used to improve pruning performance. The core formation process also uses node degree as priority, which is an approximation of the effective degree (i.e., number of selectors) used in CEDAR.

All approaches are simulated on a custom simulator. In order to generate a random network, n nodes are randomly placed in a 100×100 square region to form a unit disk graph using a range of r. For Rule k, Mesh, Tree, and CEDAR, r is set to 24. For the 2-level approach, r is 8 in the first (clustering) stage and 24 in the second (pruning) stage. For the 1-level approach, r is 6 in the first stage and 18 in the second stage. Each simulation is repeated until the 90% confidence interval is within $\pm 1\%$.

We compare performances of different approaches in terms of size of the resultant CDS. As shown in Figure 7 (a), the size of the CDS produced by Rule k increases rapidly as the network size (n) grows. The size of the DS in CEDAR is very close to the size of the CDS in Rule k, and the size of the CDS in CEDAR is much larger than in other approaches. In other words, neither Rule k nor CEDAR is very efficient in dense networks. In other approaches, the sizes of the CDS's are barely affected by the network density. For $n \geq 500$, increasing n can cause only slight difference in the sizes of the CDS's. The sizes of CDS's in those approaches depend on the number of clusterheads, which has a constant upper bound in a region with a fixed size. Among those approaches, the 1-level approach is about 20% better than the mesh approach, and the 2-level approach is about 30% better than the tree approach. Note that although the 2-level approach produces a smaller CDS than the 1-level approach, it also requires a more complicated routing scheme.

Two types of overhead are considered in our comparison: time and energy. We measure the time cost in terms of the number of rounds of message exchange. As discussed in Section Rule k completes in 2 rounds. In CEDAR, core formation requires 3 rounds, and the designation of forwarding sets needs 2 extra rounds. In other approaches, more rounds are needed to obtain a stable cluster structure. After clusterhead formation, both 1-level and 2-level approaches require two extra rounds to apply MP and Rule k. The mesh approach also requires two extra rounds: one for gathering neighboring cluster information, and another for gateway designation. When the root is pre-selected, the extra cost of the tree approach includes a flooding and at most two extra rounds for gateway notification. Figure 7 (b) shows the average number of rounds used by each approach. Rule k has the lowest cost, and the tree approach has the highest cost without root election. The 1-level, 2-level, and mesh approaches have very similar time cost. That is, compared with the mesh approach, a smaller CDS is constructed in the proposed approaches without extra time cost. The time cost of CEDAR is lower than the cost of the cluster-based approaches but higher than that of Rule k.

Considering the transmission power for different ranges (r), the energy consumption of the two proposed approaches is much lower than the other approaches. In the clustering stage of the two proposed approaches, packets are sent to a smaller range, which is only $1/3$ or $1/4$ of the normal range. A commonly used energy model [18] can be stated as $e = \alpha r^k + \beta$, where e is the energy consumption, k is usually between 2 and 4, and α, β are device specific constants. Figure 7 (c) shows the total energy consumption of each approach when $k = 2$, $\alpha = 0.001$, and $\beta = 0$. The energy



(a) Size of CDS

(b) Number of rounds

(c) Energy consumption in the sender's side.

Figure 7. Simulation results.

consumption of both proposed approaches is a fraction of the other approaches.

Overall, both proposed approaches (1) produce a smaller CDS than Rule k, CEDAR and the mesh approach, (2) a converging speed similar to that of the mesh approach, which is significantly faster than the tree approach, and (3) have significantly lower energy consumption than Rule k, CEDAR, mesh, and tree approaches.

6. Conclusions

We have proposed a novel approach to address the computation complexity issue in many local CDS construction algorithms. This approach is based on a special method of merging the clustering approach with the use of different transmission ranges. Wu and Li's marking process has been extended as an illustration of the proposed approach. Specifically, two distributed algorithms for constructing a connected backbone in MANETs have been proposed. Both analytic and simulation study confirm the effectiveness of the proposed approaches, especially in dense networks. Our future work will focus on other applications of the virtual backbone, including topology management in MANETs and point and area coverage in sensor networks.

References

[1] J. Wu and H. Li, "On calculating connected dominating set for efficient routing in ad hoc wireless networks," in *Proc. of DialM*, 1999, pp. 7–14.

[2] Y. C. Tseng, S. Y. Ni, Y. S. Chen, and J. P. Sheu, "The broadcast storm problem in a mobile ad hoc network," *Wireless Networks*, vol. 8, no. 2-3, pp. 153–167, Mar.-May 2002.

[3] A. Qayyum, L. Viennot, and A. Laouiti, "Multipoint relaying for flooding broadcast message in mobile wireless networks," in *Proc. of HICSS-35*, Jan. 2002, p. 298.

[4] J. Wu and W. Lou, "Forward-node-set-based broadcast in clustered mobile ad hoc networks," *Wireless Communications and Mobile Computing, special issue on Algorithmic, Geometric, Graph, Combinatorial, and Vector*, vol. 3, no. 2, pp. 155–173, 2003.

[5] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar, "Next century challenges: Scalable coordination in sensor networks," in *Proc. of ACM MOBICOM'99*, 1999, pp. 263–270.

[6] S. Guha and S. Khuller, "Approximation algorithms for connected dominating sets," *Algorithmica*, vol. 20, no. 4, pp. 374–387, Apr. 1998.

[7] B. Das, E. Sivakumar, and V. Bhargavan, "Routing in ad-hoc networks using a virtual backbone," in *Proc of IC3N*, Sep. 1997, pp. 1–20.

[8] K. M. Alzoubi, P. J. Wan, and O. Frieder, "New distributed algorithm for connected dominating set in wireless ad hoc networks," in *Proc. of HICSS-35*, Jan. 2002, p. 297.

[9] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris, "SPAN: an energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks," *ACM Wireless Networks Journal*, vol. 8, no. 5, pp. 481–494, 2002.

[10] F. Dai and J. Wu, "Distributed dominant pruning in ad hoc wireless networks," in *Proc. of IEEE ICC*, May 2003, vol. 1, pp. 353–357.

[11] J. Wu and F. Dai, "A generic distributed broadcast scheme in ad hoc wireless networks," in *Proc. of ICDCS*, May 2003, pp. 460–468.

[12] W. Peng and X. Lu, "On the reduction of broadcast redundancy in mobile ad hoc networks," in *Proc. of ACM MobiHoc*, June 2000, pp. 129–130.

[13] I. Stojmenovic and J. Wu, "Broadcasting and activity-scheduling in ad hoc networking," in *Ad Hoc Networking*, S. Giordano S. Basagni, M. Conti and I. Stojmenovic, Eds. IEEE Press, 2003., to appear.

[14] K. M. Alzoubi, P.-J. Wan, and O. Frieder, "Distributed heuristics for connected dominating sets in wireless ad hoc networks," *Journal of Communications and Networks*, vol. 4, no. 1, pp. 22–29, Mar. 2002.

[15] A. Ephremides, J. E. Wjeselthier, and D. J. Baker, "A design concept for reliable mobile radio networks with frequently hopping signaling," *Proceedings of IEEE*, vol. 71, no. 1, pp. 56–73, 1987.

[16] P. Sinha, R. Sivakumar, and V. Bharghavan, "Enhancing ad hoc routing with dynamic virtual infrastructures," in *Proc. of IEEE INFOCOM*, Apr. 2001, pp. 1763–1772.

[17] J. Wu and F. Dai, "A distributed formation of a virtual backbone in manets using adjustable transmission ranges," Tech. Rep., Florida Atlantic University, Nov. 2003.

[18] L. M. Feeney, "An energy-consumption model for performance analysis of routing protocols for mobile ad hoc networks," *Mobile Networks and Applications*, vol. 6, no. 3, pp. 239–249, June 2001.