Connected k-Hop Clustering in Ad Hoc Networks *

Shuhui Yang, Jie Wu
Department of Computer Science and Engineering
Florida Atlantic University
Boca Raton, FL 33431
syang1@fau.edu, jie@cse.fau.edu

Jiannong Cao
Department of Computing
Hong Kong Polytechnic University
Hung Hom, KLW, Hong Kong
csjcao@comp.polyu.edu.hk

Abstract

In wireless ad hoc networks, clustering is one of the most important approaches for many applications. A connected k-hop clustering network is formed by electing clusterheads in k-hop neighborhoods and finding gateway nodes to connect clusterheads. Therefore, the number of nodes to be flooded in broadcast related applications could be reduced. In this paper, we study the localized solution for the connectivity issue of clusterheads with less gateway nodes. We develop the adjacency-based neighbor clusterhead selection rule (A-NCR) by extending the "2.5" hops coverage theorem [17] and generalizing it to k-hop clustering. We then design the local minimum spanning tree [9] based gateway algorithm (LMSTGA), which could be applied on the adjacent clusterheads selected by A-NCR to further reduce gateway nodes. In the simulation, we study the performance of the proposed approaches, using different values for parameter k. The results show that the proposed approaches generate a connected k-hop clustering network, and reduce the number of gateway nodes effectively.

1. Introduction

The nature of wireless ad hoc networks (or simply ad hoc networks) makes them different from wireless infrastructure networks. An ad hoc network contains large numbers of hosts that communicate with each other without any centralized management. Scalability is one of the most important issues in large ad hoc networks. Clustering is an important approach to support scalability in many applications. For example, the most reliable method of information propagation in an ad hoc network is flooding, but it demands large overhead and may cause severe collision and contention. If all the hosts are organized into clusters, the information transmission flooding could be confined within each cluster. In an ad hoc network, communication overhead could be reduced by both intra-clustering and inter-clustering [6, 19]. Clustering has also been applied to routing protocols, helping to achieve smaller routing tables and fewer route updates, such as the (α, t) cluster framework [12], the B-protocol [3], and MMWN [15].

The clustering process divides the network into several clusters, and each has a clusterhead and several neighbors of this clusterhead as members. These clusters could be viewed as 1-hop clusters, in which the distance between a clusterhead and any of its members is 1 hop. There are two methods of 1-hop clustering. One is the cluster algorithm, the other is the core algorithm. The main difference between these two are whether clusterheads could be neighbors (as in core), or not (as in cluster). We focus on the first clustering method in this paper.

1-hop clustering could be extended to k-hop clustering. There are two possible extensions. The k-cluster is first defined by Krishna et al [8], in which a k-cluster is a subset of nodes that are mutually reachable by a path of at most k hops. These clusters have no clusterheads and are overlapped. The second is [7, 13], where a k-hop cluster is defined as a set of nodes within k-hop distance from a given node, their clusterhead. The difference between these two extensions is the definition of k hops, whether it is the distance between any pair of members in a cluster or the clusterhead and each member. In our paper, we use the second definition. Within the second definition, we have k-hop cluster [7, 13] (as an extension of 1-hop cluster) and k-hop core [2] (as an extension of 1-hop core). We will focus on k-hop cluster where a clusterhead forms not only a k-hop dominating set (DS) [14], where every node is in the DS or at most k hops away from the DS, but also a k-hop independent set, where clusterheads are at least $k+1$ hops away from each other. By adjusting the parameter k, the number of clusters and clusterheads could be controlled.

In ad hoc networks, clusterheads are in charge of information distribution and collection within clusters. The

*This work was supported in part by NSF grants CCR 0329741, CNS 0422762, CNS 0434533, and EIA 0130806.

communication of clusterheads or information aggregation could be accomplished in a multi-hop way. Some gateway nodes, which are non-clusterheads (members), need to be selected to connect clusterheads. To save energy and reduce signal collision, the number of these gateway nodes should be as few as possible.

In this paper, we deal with gateway selection to generate the connected k-hop clustering in ad hoc networks. The approach we use is localized, where each node performs selection based on $(2k+1)$-hop local information. To connect all the clusterheads in a localized way, we divide the process into two phases. Each clusterhead should (1) find some neighbor clusterheads first, and (2) find gateways to connect to these clusterheads. If each clusterhead is connected to every one of its neighbor clusterheads, all the clusterheads in the network are guaranteed to be connected.

We develop an adjacency-based neighbor clusterhead selection rule (A-NCR), which is an extension and generalization of Wu and Lou's "2.5" hops coverage theorem [17], for neighbor clusterhead selection in the first phase. In A-NCR, a small set of neighbor clusterheads (within $2k + 1$ hops) could be found by each clusterhead while ensuring global connectivity of clusterheads. The reduced number of neighbor clusterheads could help to result in fewer gateway nodes. In the second phase for gateway selection, we develop a local minimum spanning tree [9] based gateway algorithm (LMSTGA), which could greatly reduce the gateway nodes selected by the usual mesh-based approach. These two proposed methods could be combined as AC-LMST (where AC stands for adjacent clusterhead) to further reduce the number of resultant gateway nodes. All the approaches proposed in this paper are distributed and localized. At most $2k + 1$ hops broadcasting is needed. The parameter k is tunable, and usually small. This is because in ad hoc networks, network topology changes frequently. Therefore small k may help to construct a combinatorially stable system, in which the propagation of all topology updates is sufficiently fast to reflect the topology change.

The contributions of this paper are as follows. (1) Define the localized gateway node selection issue of connected k-hop clustering in ad hoc networks, and separate this issue into two problems of neighbor clusterhead selection and gateway selection. (2) Design an adjacency-based neighbor clusterhead selection rule (A-NCR) to address the first problem; use the local minimum spanning tree based gateway algorithm (LMSTGA) for the second problem; combine these two into AC-LMST. (3) Perform simulation to evaluate and analyze the performance of these proposed approaches.

The remainder of the paper is organized as follows: Section 2 reviews the related work in the field, including 1-hop clustering and k-hop clustering. Section 3 gives a new heuristic solution for the connected k-hop clustering, and the AC-LMST. A performance study through simulation is conducted in Section 4. The paper concludes in Section 5.

2. Related Work

Organizing a network into a hierarchical structure could make the management efficient. Clustering offers such a structure, and it suits networks with relatively large numbers of nodes. High level clustering, clustering applied recursively over clusterheads, is also feasible and effective in even larger networks.

Clustering is conducted by at first electing clusterheads; then non-clusterheads choose clusters to join and become members. As mentioned above, there are two kinds of clustering algorithms. One is the cluster algorithm [4], and the other is the core algorithm [16]. There are many ways to use different node priorities to select clusterheads. The lowest ID algorithm [10] by Lin and Gerla is widely used. In that algorithm, a node that has the lowest ID among its neighbors that have not joined any clusters will declare itself the clusterhead. Other nodes will select one of the neighboring clusterheads to join and become members. This process is repeated until every node has joined a cluster. In the lowest ID core algorithm, a node u designates the one (including itself) that has the lowest ID in u's 1-hop neighborhood as the clusterhead. Other nodes will select clusters to join. Unlike the cluster algorithm, the core algorithm runs only one round and the resultant clusterheads (also called cores) can be neighbors. Some other node priority can be used instead of node ID for the clusterhead selection, such as node degree [5], node speed, sum of distances to all neighbors, and even random timer [18].

Connectivity among clusterheads is required for most applications such as message broadcasting. Unless extra channels are used [18], all nodes are identical in power supplement; clusterheads do not connect directly with other clusterheads that are at least 2 hops away. Thus the connection between clusterheads should be accomplished in the style of a multi-hop packet relay. That is, some non-clusterheads (members) should be selected as gateway nodes to perform message forwarding between clusterheads. The distance between clusterheads of two neighbor clusters is 2 or 3 hops. One way is to select border nodes as gateways to connect the clusterheads. A *border node* is a member with neighbors in other clusters. Finding gateway nodes to connect all the clusterheads within each other's 3-hop neighborhood is another widely used method. The mesh-based scheme [16] designates a subset of members as gateways so that there is exactly one path by gateways between two neighboring clusterheads. The global tree scheme [1] minimizes the number of gateways by growing a breadth-first search tree via flooding. In [17], Wu and Lou developed the "2.5" hops coverage theorem, in which each clusterhead needs only to connect to all the clusterheads 2 hops away and some of those 3 hops away. They also designed a greedy gateway selection algorithm to connect these clusterheads that are 2.5 hops away.

There are several ways to extend the clustering to support even larger networks. One is to augment the ad hoc network with an overlay network by using a second channel, such as in [18]. The second is to incorporate multiple hierarchies to support aggregation [12]. The third is to use k-hop clustering, as in [2], [7], and [8]. When k is larger than 1, using border nodes as gateways is not enough to make clusterheads connected. Wu and Lou's greedy gateway selection algorithm is not suitable either. One solution is to use a centralized approach to construct a global minimum spanning tree to connect all the clusterheads. Another approach is to use some existing routing algorithms to send messages among clusterheads [5]. To our best knowledge, there is no localized gateway selection algorithm in k-hop clustering networks thus far. Some other detailed information about clustering could be found in [13].

3. A Heuristic Solution for Connected k-Hop Clustering

We use the traditional lowest ID clustering algorithm, and apply it to a k-hop neighborhood. We denote the original connected network after clustering, which has selected clusterheads and classified members, as G. In the clustering algorithm, nodes that have the highest priority within their k-hop neighborhood (including only nodes that have not joined any clusters) declare themselves as clusterheads, and broadcast the clusterhead declaration messages in this neighborhood. Each non-clusterhead collects broadcast messages and selects one cluster to join as a member. Then the same procedure is carried out among nodes that have not joined clusters iteratively until every node joins a cluster. For a non-clusterhead that has received more than one clusterhead declaration message within its k-hop neighborhood, there are several ways for it to decide which cluster to join. (1) ID-based: the node will select the clusterhead with the smallest ID as its clusterhead. (2) Distance-based: the node will select the nearest clusterhead as its clusterhead. (3) Size-based: the decision is made considering the balance of size of clusters. Since each non-clusterhead node selects only one cluster to join, the k-hop clustering algorithm generates non-overlapped clusters.

A *cluster graph* $G^{'}$ is defined as follows.

Definition 1 $G^{'} =< V, E >$. *V is the set of clusterheads; each unidirectional link e ($e \in E$) between node u and v ($u, v \in V$) indicates a path connecting u and v, which consists of gateways only.*

Therefore, our goal is to find a connected $G^{'}$, using as few gateways as possible by a localized solution. We separate it into two phases.

1. **Neighbor clusterhead selection.** Each clusterhead collects information of other clusterheads within its local neighborhood, and designates all/some of them as its neighbor clusterheads. The connectivity of clusterheads should be guaranteed as long as each clusterhead is connected to every one of its neighbor clusterheads.

2. **Gateway selection.** Each clusterhead finds gateways to connect to all its neighbor clusterheads. It can find only gateways to directly connect to some of them, but globally, all the clusterheads are connected.

The challenge in neighbor clusterhead selection is to select as few clusterheads as possible locally, but if each clusterhead finds gateways to collect to every one of its neighbor clusterheads, all clusterheads are connected globally.

3.1. Adjacency-Based Neighbor Clusterhead Selection

Usually, each clusterhead tries to connect to all neighbor clusterheads within $2k + 1$ hops in k-hop clustering to ensure global connectivity among clusterheads. The main result in this subsection is that only a special subset of neighbor clusterheads called adjacent clusterheads needs to be connected to ensure connectivity. *Adjacent clusters* means that there are two neighbor nodes, with one from each cluster. Accordingly, *adjacent clusterheads* are the two clusterheads of adjacent clusters. It is easy to see that in k-hop clustering network G, the distance between every two adjacent clusterheads is m, where $k + 1 \leq m \leq 2k + 1$. If we use sets C_1 and C_2 to denote two clusters, the formal definition of adjacent clusters is as follows.

Definition 2 *Clusters C_1 and C_2 are adjacent clusters if and only if there exist $w_1 \in C_1$, $w_2 \in C_2$, and w_1, w_2 are neighbors in the network G. (w_1, w_2 can be clusterhead, but not both.)*

According to the concept of adjacent clusters, the adjacent cluster graph $G^{''}$ is defined as follows.

Definition 3 $G^{''} =< V, E' >$. *V is the set of clusterheads; each link e ($e \in E'$) between nodes u and v ($u, v \in V$) indicates the two clusters with heads u and v are adjacent clusters.*

Theorem 1 *The adjacent cluster graph $G^{''}$ is connected.*

Proof: Because of the connectivity of graph G, for every pair of vertices u and v in $G^{''}$, which are clusterheads, there exists a path in G to connect them. We denote the path

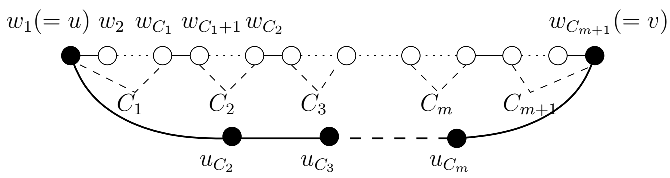


Figure 1. Proof of Theorem 1.

as $w_1(=u), w_2, \ldots, w_{C_1}, w_{C_1+1}, \ldots, w_{C_2}, \ldots, w_{C_{m-1}+1}, \ldots, w_{C_m}, w_{C_m+1}, \ldots, w_{C_{m+1}}(=v)$. The nodes on the path belong to different clusters. We use $C_1, C_2, \ldots, C_m, C_{m+1}$ to denote the clusters, and $u_{C_1}, u_{C_2}, \ldots, u_{C_m}, u_{C_{m+1}}$ as clusterheads of these clusters in sequence, as Figure 1. Nodes w_{C_1} and w_{C_1+1} are neighbors, thus u and u_{C_2} are adjacent clusterheads, and there is a link in $G^{''}$ between them. In the same way, u_{C_2} is connected to u_{C_3}, u_{C_3} to $u_{C_4}, \ldots$, and u_{C_m} to v. Therefore, u and v are connected. $G^{''}$ is connected. □

Note that a simple and intuitive way to connect all the clusterheads in G is for each clusterhead to select all the clusterheads within $2k+1$ hops as its neighbor clusterheads and find gateways to connect itself and each of them, such as [16] does. We can see that the cluster graph constructed by this simple method is a super graph of $G^{''}$, therefore it is not efficient enough.

Wu and Lou proposed the 2.5 hops notion for the clusterheads connection [17] when k is 1. That is, clusterheads are connected by carefully selecting non-clusterhead nodes locally at each clusterhead to connect clusterheads within its 2.5 hops. They use the notion of 2.5 hops coverage, where each clusterhead covers clusterheads within its 2 hops neighborhood, and clusterheads within 3 hops that have members within its 2 hop neighborhood. Note that when k is 1, distance between two adjacent clusterheads is either 2 or 3.

Figure 2 is an illustration of their proposed 2.5 hops coverage theorem. (a) is the graph after clustering. (b) uses the simple method to find neighbor when $k = 1$, that is to connect all the clusterheads within 3 hops. (c) is the 2.5 hop connection theory, which can reduce the connections, such as link 2 to 4, 4 to 2, and 1 to 4. Therefore, some unidirectional connections may exist. There are still some redundant connections, such as link 4 to 1, which could be removed. We can see that the directional cluster graph generated by this 2.5 hops coverage theorem is still a super graph of adjacent cluster graph $G^{''}$. Therefore, it is still not efficient enough, and could be extended to further remove redundant connections among clusterheads, and generalized to k-hop clustering.

Based on Theorem 1, we develop the following neighbor clusterhead selection rule, which is to select only adjacent clusterheads, not all the $2k+1$ neighbor clusterheads to connect, to reduce redundant connections.

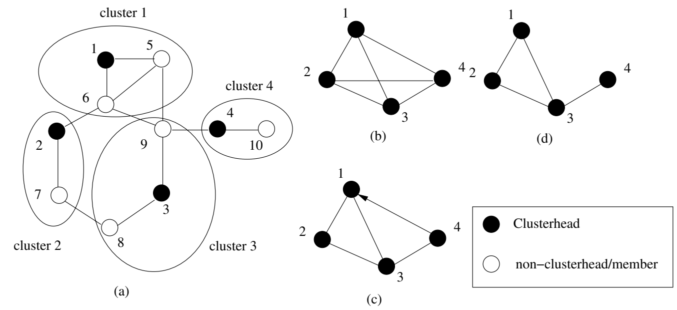


Figure 2. (a) the network G, (b) 3 hops connection, (c) 2.5 hops connection, (d) adjacency-based connection ($G^{''}$) ($k = 1$).

Adjacency-Based Neighbor Clusterhead Selection Rule (A-NCR): *In a k-hop network G which is already clustered, each clusterhead selects the adjacent clusterheads within $2k+1$ hops as its neighbor clusterheads to connect.*

The cluster graph $G^{'}$ constructed by A-NCR is exactly the adjacent cluster graph $G^{''}$, and therefore is efficient.

Wu and Lou's 2.5 hops coverage is a special case of A-NCR, when k is 1. Because if cluster C_2 with clusterhead v has no member within 2 hops of clusterhead u of cluster C_1, these two clusters must be separated by a node, which belongs to neither C_1 nor C_2. We can see that as a result of our method, all the remaining connections between clusterheads are symmetric, therefore the cluster graph $G^{'}$ is still undirected.

3.2. LMST-Based Gateway Algorithm (LMSTGA)

Generally, there are several ways to connect all the clusterheads to form a connected graph. Globally, a minimum spanning tree could be constructed, connecting all the clusterheads via gateways. Note that we use hops between two nodes as the distance separating them. Locally, each clusterhead could find a shortest path to connect to every one of its adjacent clusterheads, and uses the non-clusterheads on the path as gateways. To further reduce the number of gateways, we apply a local minimum spanning tree (LMST) [9] algorithm for connecting to adjacent clusterheads.

Li, Hou, and Sha devised a distributed and localized algorithm (LMST) for the topology control problem starting from a minimum spanning tree. In the network, each node builds its local MST independently based on the location information of its 1-hop neighbors and only keeps links to 1-hop nodes on its local MST in the final topology. The algorithm produces a connected topology. That is, all the links marked together with all the nodes can form a connected

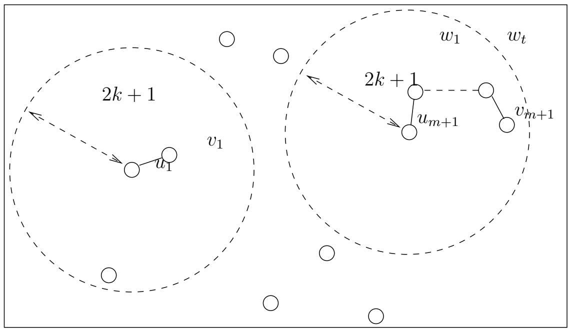


Figure 3. Proof of LMSTGA in $G^{'}$, using A-NCR to select neighbor clusterheads.

graph. An optional phase is provided where the topology is transformed to one with bidirectionl links.

We extend the LMST algorithm, and apply it to our gateway finding procedure. In our extension, we first create "virtual links" among clusterheads. We find a shortest path between every pair of clusterheads if they are adjacent neighbors, and have this path as their "virtual link", using hop count to represent each clusterhead's pairwise "virtual distance". Therefore a global "virtual graph" is formed containing clusterheads and all the virtual links. This virtual graph is the same as the adjacent graph $G^{''}$. For each clusterhead, all its adjacent clusterheads are viewed as in its virtual "1-hop" neighborhood, although they may be up to $2k+1$ hops away in G. Then, the LMST algorithm is applied on this clusterhead. The IDs of two nodes of a virtual link can be used to break a tie in hop count if needed. When a virtual link is selected in LMST, all nodes on the virtual link are selected as gateways.

LMST-Based gateway algorithm (LMSTGA) : *In a k-hop adjacent cluster graph $G^{''}$, each clusterhead finds a shortest path to every one of its adjacent clusterheads, and marks it as a virtual link, using hop count as the virtual distance. Each clusterhead constructs a local minimum spanning tree (LMST) among all the adjacent clusterheads rooted at itself, using only virtual links. Then each clusterhead selects the on-tree neighbors (i.e. neighbors on the LMST) to connect to by marking all the intermediate nodes as gateways on the selected virtual links to these neighbors.*

The proof of the following theorem is similar to the one that proves the connectivity of LMST in [9], except for the concept of virtual link and "1-hop" neighborhood.

Theorem 2 *The clusterheads, gateway nodes selected by LMSTGA, using A-NCR to select neighbor clusterheads, and the links among them in the given network G form a connected graph.*

Proof: We have proved that if each clusterhead could be connected to every one of its adjacent clusterheads, the whole cluster graph $G^{'}$ is connected. Therefore, we only need to prove that after applying LMSTGA, there is a path between each clusterhead and every one of its neighbors.

We assume the virtual link between clusterheads u and v denoted as $u \leftrightarrow v$, is selected by LMSTGA, and the virtual distance of the virtual link is $d(u, v)$; if there is a path formed by selected virtual links to connect u and v, say $u \leftrightarrow w_1 \leftrightarrow \ldots \leftrightarrow w_t \leftrightarrow v$, we use $u \Leftrightarrow v$ to denote it. Let us sort all the virtual links in $G^{''}$, say (u_1, v_1), (u_2, v_2), ..., (u_z, v_z), by increasing distance. We can have the sequence: $d(u_1, v_1) < d(u_2, v_2) < \ldots < d(u_z, v_z)$. Then we use induction to prove that every one of these pairs is connected by those virtual links selected by LMSTGA.

Basic: $m = 1$. Since $d(u_1, v_1)$ is the smallest link in the whole graph, it must be the first virtual link selected by u_1. Therefore, we could have $u_1 \leftrightarrow v_1$.

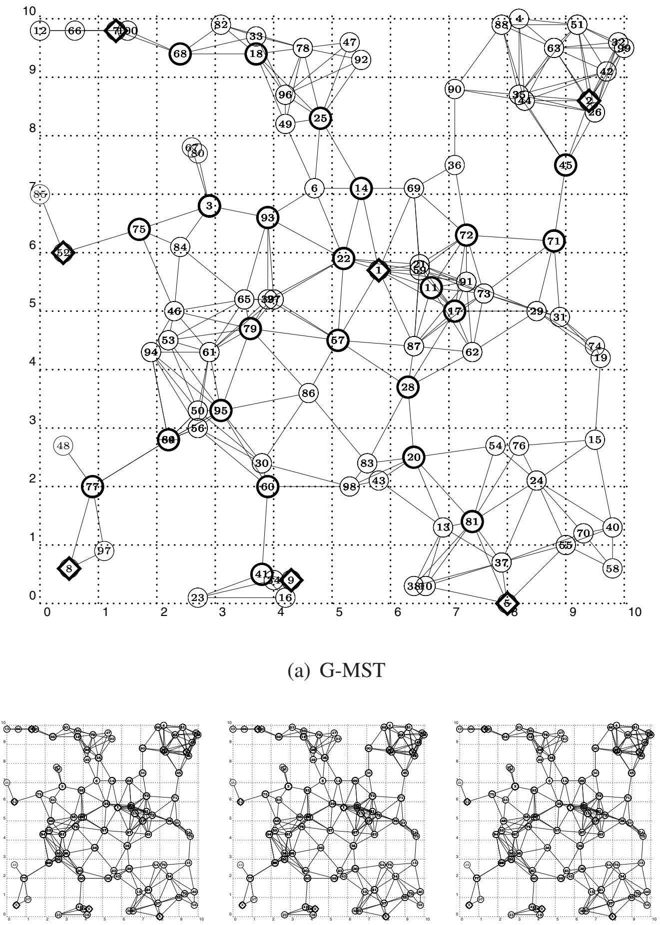
Induction: we assume $u_i \Leftrightarrow v_i$, $i = 1, \ldots m$. Now we prove $u_{m+1} \Leftrightarrow v_{m+1}$. (1) Suppose $u_{m+1} \leftrightarrow v_{m+1}$. That is to say, the virtual link between them is selected by LMSTGA, and $u_{m+1} \Leftrightarrow v_{m+1}$. (2) Suppose a virtual link between them is not selected by LMSTGA. See Figure 3, clusterhead u_{m+1} does not select the virtual link between v_{m+1} and itself. On the LMST rooted at u_{m+1}, there must exist a path to v_{m+1}, say u_{m+1}, w_1, ..., w_t, v_{m+1}. Every virtual link on this path is smaller than $d(u_{m+1}, v_{m+1})$, otherwise it could be replaced by link (u_{m+1}, v_{m+1}) to reduce the weight of the MST. Since we already assume that the virtual link smaller than (u_{m+1}, v_{m+1}) is connected, we could have $u_{m+1} \Leftrightarrow w_1 \Leftrightarrow, \ldots, \Leftrightarrow w_t \Leftrightarrow v_{m+1}$. Therefore, we have $u_{m+1} \Leftrightarrow v_{m+1}$. $\square$

LMSTGA combined with A-NCR is to apply LMST selection on adjacent clusterheads, which is denoted as AC-LMST. The following algorithm is executed on clusterhead u, assuming k-hop clustering has been accomplished.

Algorithm AC-LMST (u)

broadcast within $2k+1$ hops
collect broadcast messages
use A-NCR to find neighbor clusterhead set S
for all $i \in S$
    find a shortest path p_i to i
    designate hop count c_i of p_i as its distance
broadcast set S and distance to every one in S
collect broadcast information
construct an LMST among nodes in S rooted at u
for all $i \in S$ which are also on-tree neighbors
    set nodes on p_i as gateway nodes

End_AC-LMST.



(a) G-MST

(b) NC-Mesh (c) NC-LMST (d) AC-LMST

Figure 4. Example of gateway selection using different algorithms ($N = 100$, $D = 6$, $k = 2$).

Figure 4 is an example of resultant cluster graphs from different gateway selection algorithms. There are 100 nodes in the original network. Lowest ID clustering and ID-based member classification are used. The average degree of each node is 6. k is 3. There are 7 clusterheads (marked by diamonds). (a) shows the result by global minimum spanning tree method (G-MST). There are 23 gateways (bold circles), and they together with the clusterheads form a MST. (b) shows the mesh-based method applied on all the neighbor clusterheads within $2k + 1$ hops (NC-Mesh). There are 35 gateways. (c) is the LMST-based method, applied on all neighbor clusterheads within $2k + 1$ hops (NC-LMST). There are 28 gateways. (d) is AC-LMST, which is LMST applied on adjacent clusterheads. There are 26 gateways.

3.3. Discussion

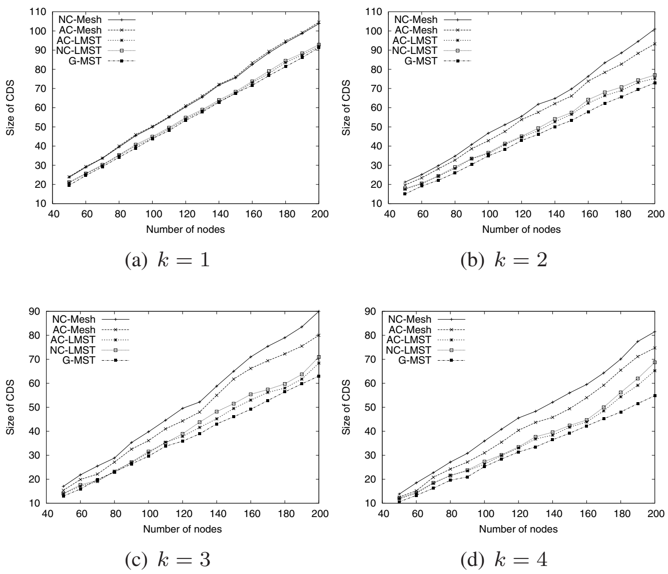
One of the special characteristics of ad hoc networks is its restricted power supply. Clustering protocols should be oriented towards power-saving and energy-efficiency. One way for power-aware design is to rotate the role of clusterhead to prolong the average lifespan of each node, assuming that a clusterhead consumes more energy than a regular node. Therefore, residual energy level instead of lowest ID can be used as node priority in the clustering process.

k-hop clustering can also easily handle the dynamic situation due to node movement and node switch-on/off operations. Consider a situation when a node "disappears": if it is non-clusterhead and non-gateway, nothing needs to be done with respect to the existing CDS; if it is non-clusterhead but gateway, only the corresponding clusterhead needs to re-run the gateway selection process (to have a local fix); if it is a clusterhead, the clusterhead selection process is applied. Since the number of clusterheads is relatively small, especially for a relatively large constant k, the chance of re-applying the clusterhead selection process is also small.

4. Simulation

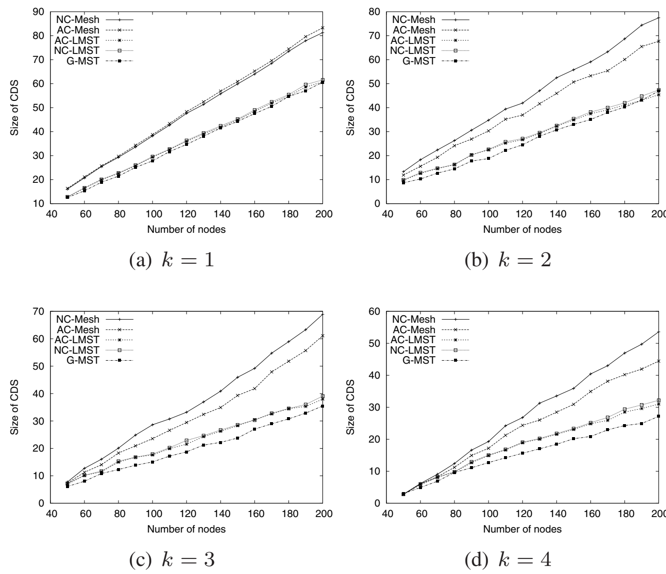
This section presents results from our simulation study. The efficiency of the proposed approaches are evaluated and compared with existing ones. All approaches are simulated on a custom simulator, which simulates the k-hop clustering algorithm. For the gateway selection approaches, it simulates neighbor clusterheads (NC) selection, adjacent clusterheads (AC) selection, mesh-based gateway, LMST-based gateway, and also global minimum spanning tree (G-MST) based gateway. Therefore there are four algorithms, NC-Mesh, AC-Mesh, NC-LMST, AC-LMST, in addition to G-MST to be compared. We use G-MST as a lower bound. In fact, G-MST has a constant approximation ratio to the optimal k-hop CDS for a constant k. In 1-hop clustering, the clusterheads and gateway nodes will form a connected dominating set (CDS) to carry out data propagation. Finding a minimum CDS (MCDS) is an NP-complete problem [11]. Clusterheads together with gateways generated by k-hop clustering form a k-hop CDS. Finding a minimum k-hop CDS is also NP-complete.

To generate a random ad hoc network, N nodes are randomly placed in a restricted 100×100 area. In the ad hoc network, we assume all nodes have the same transmission range. We will ignore practical details such as collision and contention, assuming that an ideal MAC layer protocol will take care of them. The tunable parameters in our simulation are as follows. (1) The node number N. We change the number of deployed nodes from 50 to 200 to see the scalability of the algorithms. (2) The average node degree D. We use 6 and 10 as average node degree to see the effect of



(a) $k = 1$ (b) $k = 2$ (c) $k = 3$ (d) $k = 4$

Figure 5. Comparison of different algorithms in sparse networks ($D = 6$).



(a) $k = 1$ (b) $k = 2$ (c) $k = 3$ (d) $k = 4$

Figure 6. Comparison of different algorithms in dense networks ($D = 10$).

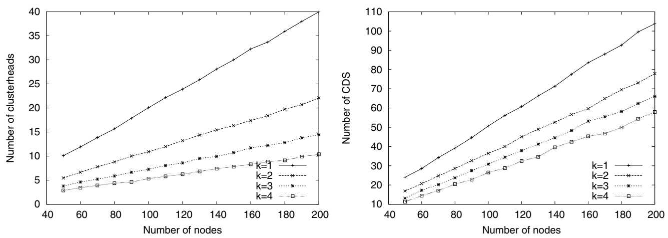
link density on the algorithms. (3) The clustering parameter k. k controls the confines of each cluster, and the number of clusterheads. We use 1, 2, 3 and 4 as its value. The metrics we used to measure the performance of the algorithms are the number of gateway nodes selected, together with clusterheads, and the size of the k-hop CDS. For each tunable parameter, the simulation is repeated 100 times or until the confidence interval is sufficiently small ($\pm 1\%$, for the confidence level of 90%).

Figure 5 is the comparison of the different gateway selection algorithms with average node degree of 6, that is, each node has around 6 nodes as its neighbors, which will result in a relatively sparse graph. Four algorithms are compared. The first is NC-Mesh. The second is AC-Mesh. The third is NC-LMST. In this approach, the LMST algorithm is applied on all the clusterheads within $2k + 1$ hops of the current clusterhead. The last one is AC-LMST, which combines the LMST and A-NCR approaches to make the most of them. We can see that the number of gateway nodes selected is proportional to the number of nodes in the network, and all the approaches have the property of scalability. (a) is for $k = 1$. We can see that AC-Mesh has little advantage over NC-Mesh, as does AC-LMST over NC-LMST. The method of LMST can reduce gateway nodes of Mesh by over 10%. (b) $\sim$ (d) are for $k = 2, 3, 4$. When k is greater than 1, A-NCR works. AC-Mesh reduces gateway nodes of NC-Mesh and AC-LMST reduces gateways of NC-LMST as well. But from the simulation, we can see that the LMST-based approach is more effective than A-NCR. AC-LMST is the most effective one.

Figure 6 is the comparison of these algorithms with average node degree of 10, which will result in a relatively dense graph. Compared with Figure 5, the number of clusterheads and gateway nodes is smaller here. The performance of the four algorithms is similar to that of Figure 5, except that the advantage of AC-LMST over NC-LMST is even less.

Figure 7 shows the effect of the clustering parameter k. AC-LMST is used to find gateways. (a) is the number of clusterheads using different k. The larger the k, the fewer the clusterheads, thus the clusters. (b) is the size of CDS, the number of clusterheads together with gateways. We can see that the size of the resultant CDS becomes smaller with the increase of k, although the number of gateways becomes larger.

Simulation results can be summarized as follows. (1) The proposed A-NCR reduces the number of gateway nodes. (2) The AC-LMST which is a combination of A-NCR and extended LMST could further reduce the number of gateway nodes. (3) The proposed approaches are scalable and suited for both sparse and dense networks. (4) Of these two approaches, LMST is more effective than A-NCR, and AC-LMST has little performance improvement of LMST, especially in dense networks. (5) Larger k results in fewer clusterheads and more gateways, but all together, the size of the final CDS is smaller. (6) AC-LMST has very close performance to G-MST, which is used as a low bound for the number of gateways selected.



(a) Number of clusterheads (b) Number of nodes in CDS

Figure 7. Comparisons with different k, using LMSTGA ($D = 6$).

5. Conclusion

In this paper, we study the issue of connected k-hop clustering. We separate this problem into two steps in the localized solution. One is neighbor clusterhead selection, the other is gateway node selection. We extend and generalize the 2.5 hops coverage theorem to reduce the neighbor clusterheads to be connected, and develop the A-NCR approach. For the second phase, we extend the LMST algorithm to apply it on the virtual graph abstracted from the given network to select a small number of gateway nodes. AC-LMST is the use of both approaches to find gateway nodes. These two proposed methods could be used separately or together. In the future, we will design a movement-sensitive maintenance policy for the gateway selection algorithm. Communication overhead increases with the growth of the value of k. We will perform some in-depth simulation which should help in analyzing the tradeoff between communication overhead and efficiency of k-hop.

References

[1] K. Alzoubi, P. J. Wan, and O. Frieder. New distributed algorithm for connected dominating set in wireless ad hoc networks. In *Proc. of HICSS*, page 297, 2002.

[2] A. D. Amis, R. Prakash, T. H. P. Vuong, and D. T. Huynh. Max-min d-cluster formation in wireless ad hoc networks. In *Proc. of IEEE Infocom*, 2000.

[3] S. Basagni, D. Turgut, and S. K. Das. Mobility-adaptive protocols for managing large ad hoc networks. In *Proc. of the IEEE International Conference on Communications (ICC)*, page June, 2001.

[4] A. Ephremides, J. E. Wieselthier, and D. J. Baker. A design concept for reliable mobile radio networks with frequency hoping signaling. In *Proc. of IEEE*, pages 56–73, 1987.

[5] M. Gerla and J. T. C. Tsai. Multicluster, mobile, multimedia radio network. *Wireless Networks*, 15(7):255C265, 1995.

[6] V. Kawadia and P. R. Kumar. Power control and clustering in ad hoc networks. In *Proc. of IEEE Infocom*, 2003.

[7] D. Kim, S. Ha, and Y. Choi. k-hop cluster-based dynamic source routing in wireless ad-hoc packet radio networks. In *Proc. of IEEE VTC*, page 224C228, 1998.

[8] P. Krishna, N. N. Vaidya, M. Chatterjee, and D. K. Pradhan. A cluster-based approach for routing in dynamic networks. *ACM SIGCOMM Computer Communication Review*, 49:49–64, 1997.

[9] N. Li, J. Hou, and L. Sha. Design and analysis of an MST-based topology control algorithm. In *Proc. of IEEE Infocom*, 2003.

[10] C. R. Lin and M. Gerla. Adaptive clustering for mobile wireless networks. *IEEE Journal on Selected Areas in Communications*, 15(7):1265C1275, 1997.

[11] M. V. Marathe, H. Breu, H. B. Hunt III, S. S. Ravi, and D. J. Rosenkrantz. Simple heuristics for unit disk graphs. *Networks*, 25:59–68, 1995.

[12] A. B. McDonald and T. F. Znati. A mobility-based framework for adaptive clustering in wireless ad hoc networks. *IEEE Journal on Selected Areas in Communications*, 17(8):1466–87, August 1999.

[13] F. G. Nocetti, J. S. Gonzalez, and I. Stojmenovic. Connectivity based k-hop clustering in wireless networks. *Telecommunication Systems*, 22:1C4, 205C220, 2003.

[14] A. Qayyum, L. Viennot, and A. Laouiti. Multipoint relaying for flooding broadcast message in mobile wireless networks. *Proc. of the 35th Hawaii International Conference on System Science (HICSS-35).* 2002, (CD-ROM).

[15] R. Ramanathan and M. Steenstrup. Hierarchically-organized, multihop mobile wireless networks for quality-of-service support. *Mobile Networks and Applications*, 3(1):101–19, 1998.

[16] P. Sinha, R. Sivakumar, and V. Bharghavan. Enhancing ad hoc routing with dynamic virtual infrastructures. In *Proc. of IEEE Infocom*, 2001.

[17] J. Wu and W. Lou. Forward node set based broadcast in clustered mobile ad hoc networks. *Wireless Communications and Mobile Computing , a special issue on Algorithmic, Geometric, Graph, Combinatorial, and Vector Aspects of Wireless Networks and Mobile Computing*, 3(2):141–154, March 2003.

[18] K. Xu, X. Hong, and M. Gerla. An ad hoc network with mobile backbones. In *Proc. of the IEEE International Conference on Communications (ICC)*, 2002.

[19] O. Younis and S. Fahmy. Distributed clustering in ad-hoc sensor networks: A hybrid, energy-efficient approach. In *Proc. of IEEE Infocom*, 2004.

0190-3918/05 $20.00 © 2005 **IEEE**