The Impact of Clustering in Distributed Topology Control [1]

Liang Zhao [2,3] and Errol L. Lloyd [2]

***Abstract*— Topology control is the problem of assigning power levels to the nodes of an ad hoc network so as to maintain a specified network topology while minimizing energy consumption (either minimizing the maximum power used by any node or minimizing the total (i.e. average) power used by the nodes). In [18], a hybrid framework for distributed topology control based on clustering was proposed. That framework, called *CLTC*, specifies algorithms for both 1-connected and 2-connected topologies, and works with any clustering algorithm. *CLTC* utilizes centralized topology control within each cluster, but is otherwise fully distributed, hence the characterization of the method as *hybrid*.**

This paper studies the effect of six representative clustering methods on the quality of the topology control solutions provided by *CLTC*. The results establish that the most important factors in determining the performance of *CLTC* are the average cluster size and the closeness of nodes in clusters. This leads to a tradeoff between the energy consumption, the complexity of cluster formation, and the scope to which the operations of *CLTC* are fully distributed. The paper also shows that, in general, there is a considerable increase in power usage (in the vicinity of 150%) by requiring a 2-connected network versus a 1-connected network.

***Index Terms*— Ad Hoc Network, Distributed Clustering, and Performance Evaluation.**

I. Introduction

In an ad hoc network, where nodes are deployed without any wired infrastructure and communicate via multihop wireless links, the *network topology* is autonomously formed based on the nodes' locations and transmission ranges. The nodes communicate through wireless links, with each node acting as a relay when necessary to allow multihop communications.

The performance of the network can be impacted in a major way by the network topology. A dense topology may induce high interference, which in turn reduces the effective network capacity due to limited spatial reuse and may cause unnecessarily high energy consumption. In contrast, a sparse topology is vulnerable to network partitioning due to node or link failures.

Topology control for ad hoc networks aims to maintain a *specified topology*, such as requiring that the network be 1-connected[4] or that the network be 2-connected. The desired effect of topology control is to reduce energy consumption, reduce MAC layer interference between adjacent nodes, and to increase the effective network capacity.

The primary method of accomplishing topology control is by adjusting the transmission powers of the network nodes. That is, each node is assigned a transmission power level so that the induced graph of the network satisfies the specified topology. Further, the assignment of transmission powers to nodes aims to optimize some function of those powers.

Over the past several years, the study of topology control in ad hoc networks has encompassed both theoretical and applied issues. Typically, the theoretical papers have focused on centralized algorithms and provable performance bounds, while the more applied papers have considered distributed algorithms evaluated by means of simulations. The work in [18] described a *hybrid* topology control *framework* named *Cluster based Topology Control (CLTC)* aimed at taking advantage of the centralized algorithms in a distributed setting. The key points of *CLTC* will be introduced in subsection II-B. In *CLTC*, *any* clustering algorithm can be utilized in forming the clusters. The quality of the solution produced by *CLTC* is directly dependent on the particular method used for clustering. In this paper we provide a comparative study of the performance of the *CLTC* approach using six clustering methods. The effect of these clustering methods on the performance of *CLTC* is studied for both 1 and 2-connected topologies.

II. Background

A. Model and Objectives

Our ad hoc network model is based on one described in [17]. In this model, for each ordered pair (u, v) of transceivers, there is a *transmission power threshold*, denoted by $p(u, v)$, where a signal transmitted by the transceiver u can be received by v only when the transmission power of u is at least $p(u, v)$. The transmission power threshold for a pair of transceivers depends on the distance between the transceivers, the direction of the antenna at the sender, interference, noise, etc. [17]. In this paper we utilize only *symmetric* thresholds, where $p(v, u) = p(u, v)$.

[1]Prepared through collaborative participation in the Communications and Networks Consortium sponsored by the U. S. Army Research Laboratory under the Collaborative Technology Alliance Program, Cooperative Agreement DAAD19-01-2-0011. The U. S. Government is authorized to reproduce and distribute reprints for Government purposes not withstanding any copyright notation thereon.

[2]Computer and Information Sciences, University of Delaware, Newark, DE 19716. {zhaol, elloyd}@cis.udel.edu. Phone: 302-831-{1949, 1958}. Fax: 302-831-8458.

[3]The author presenting the paper.

[4]This is commonly referred to simply as *connected*. In this paper we use *1-connected* to avoid any confusion with *2-connected*.

As noted earlier, topology control algorithms achieve a given topology while optimizing some function of the transmission powers. The two optimization functions most commonly utilized are:

- minimize the maximum power used by any network node. We refer to this criterion as MINMAX.
- minimize the total power used by all of the nodes in the network. This is equivalent to minimizing the average power used by the nodes. We refer to this criterion as MINTOTAL.

In regard to the topology of the graph induced by the power assignments to the network nodes, the topologies of most interest are those related to the network connectivity, namely 1-connected and 2-connected. Here, 1-connected simply means that the resulting network is connected, while 2-connected means that the resulting network can be disconnected only with the removal of at least two nodes.

B. What is CLTC?

In this subsection, we briefly review the *CLTC* framework [18]. We begin by noting that *CLTC* does not require the global topology to be known by any entity. Rather, the framework relies on *clustering* where nodes autonomously form groups (clusters) and select a clusterhead for each cluster. The framework consists of three phases. The reader is referred to [18] for a full description.

- **Phase 1 - Form clusters:** In the first phase clusters are formed and clusterheads are selected. The clusterheads assume the primary responsibility for the subsequent two phases. Note that the operations in phases 2 and 3 are independent of the specific clustering algorithm utilized in this first phase. The clustering methods studied in this paper are described in Section III.
- **Phase 2 - Intra-cluster Topology Control:** In this phase, each clusterhead obtains the coordinates of each node in its cluster, and then utilizes an appropriate centralized algorithm to calculate the power assignments for all members of its cluster, such that the resulting cluster topology satisfies the given connectivity constraint (i.e. 1 or 2-connected). Note that some clusters may not be able to achieve a topology with the specified connectivity, even if all nodes transmit at full power. For these *weak clusters* all nodes are set to transmit at full power, and it is the responsibility of Phase 3 to insure that the network as a whole achieves the specified connectivity. Clusters that are able to achieve the specified connectivity are termed *strong clusters*.
- **Phase 3 -Inter-cluster Topology Control:** Connectivity between adjacent clusters (and hence for the network as a whole) is determined here. Throughout this phase, we refer to k connectivity, where k is either 1 or 2, depending on the specified topology. The phase consists of two steps.
  1) *Connecting adjacent strong clusters:* For each pair of adjacent strong clusters this step ensures that there are k disjoint links between those clusters provided k such links exist. For a given strong cluster A, this determination relies on information from each strong cluster B that contains at least one node adjacent to a node of A. Using this information, the clusterhead for A (likewise, the clusterhead for B) executes a connectivity specific algorithm that locates k disjoint links of minimum total power between A and B. When there don't exist k disjoint links between A and B, the clusterhead for A simply assigns full power to each of its members who are adjacent to a node in B, thus preserving all of the links between clusters A and B. After computing the links to each adjacent strong cluster, the clusterhead performs an optimization to remove certain redundant links.
  2) *Handling weak clusters:* In this step, each node x that belongs to a weak cluster will inform each node y within its transmission range that y should utilize their maximum transmission power in order to preserve x's original connectivity. It was shown in [18] that this is sufficient to insure the k-connectivity of the network as a whole in the presence of weak clusters.

After Phase 3 is completed, each node is assigned a transmission power that is the largest of the transmission powers assigned to the node in Phase 2, Phase 3 step 1, and Phase 3 step 2.

III. CLUSTERING METHODS

Clustering methods typically form clusters either on the basis of node weights (such as node id, node degree, and combined weight), or on the basis of a particular graph property (i.e., clustering by dominating set, clique, or spanning tree). In addition, clustering techniques differ in regard to the scope of the local information that is collected (e.g. 1-hop or 2-hop), and in how the clusterheads are selected. In this paper, we utilize methods representative of a cross section of approaches to clustering. Six disparate clustering algorithms are studied thus providing a comparison of the effect of various clustering techniques on the performance of *CLTC*.

A. Weight based Clustering

In weight based clustering, every node of the cluster computes a weight. These weights differ among algorithms and may include one or more of the following: node id, node degree, node load, link states, node speed, battery power, etc. In general, the elements that constitute that weight are selected to meet some practical requirements while guaranteeing the network is connected.

Some weight based clustering methods have been proposed, such as [2], [11]. In this paper, we consider three representative weight based clustering methods:

- *DDCA* as described in [14], is a clustering method based on (α, t) criteria. The (α, t) criteria indicates that in a cluster the link between the clusterhead and any of its cluster members must be available over time period t with a probability α regardless of the hop distance between them. Details are found in [14].
- *ADB* as described in [8] and utilized in [18], is a general method that uses a complicated notion of the *height* of a node to determine the clusterheads. However, in the case of stationary nodes, the use of height is equivalent to clusterheads being determined as follows: Each node X waits until all of the nodes of smaller weight in its 2-hop neighborhood have decided if they are a clusterhead. If there *is* at least one clusterhead among those nodes, then node X joins the cluster of the node of smallest weight. Otherwise, node X declares itself to be a clusterhead.
- *WCA* is a clustering method where each node determines its weight based on four values, only two of which apply to this study where nodes are not mobile and each node is assumed to begin with the same battery power. The first of those two applicable values is d_v, the *degree difference*, which is the absolute value of the difference between the degree of v if it transmits at full power and a specified threshold value (the value 2 is recommended in [3]). The second applicable value is D_v, the sum of the distances between v and its full power neighbors. These values are combined to yield a node weight of $W_v = w_1 d_v + w_2 D_v$, where w_1 and w_2 are weighting factors. In this study we let w_1 and w_2 have the values 0.7 and 0.2 respectively, as suggested in [3].
  Clusterheads in WCA are self-identified by having the lowest weight within their one hop neighborhood. Subsequently, nodes join a cluster where the clusterhead is within one hop and choose between several such possibilities by selecting the closest clusterhead in distance.

B. Graph based Clustering

Among graph based clustering methods, the predominant approach is to utilize dominating sets (such as [4], [7]). In this paper, we study two graph based methods.

- *MMD* is a clustering method, where the clusterheads form a d-hop dominating set. In this paper, the value of d is set to 2. One goal of MMD is to minimize the number of clusterheads. Another goal is to evenly distribute the nodes among the clusterheads. In MMD, each node exchanges the information of the largest ID in its d-hop neighborhood (called floodmax in [1]), then redo it once more but for the smallest ID (called floodmin). Based on the collected information, MMD using the specified rules in [1] selects the clusterheads.
- *Clique clustering* is a method where each node of the cluster can communicate directly with every other node of the cluster, hence each cluster is a *clique* when nodes transmit at maximum power [10]. This forces the cluster nodes to be in relatively close physical proximity to one another.
  Clique clustering has strong theoretical foundations and seems having appealing features in regard to both the intra-cluster and inter-cluster phases of *CLTC*.

IV. Testbed Specification

To compare the performance of the selected clustering methods in the *CLTC* framework, experiments were conducted by placing a specified number of nodes in a 4 mile by 4 mile area, using a uniform random distribution. Four network sizes were studied: 50 nodes, 100 nodes, 150 nodes and 200 nodes. For each number of nodes, ten trials were generated and all of the numbers that we report are averages over those ten trials.

The testbed was developed directly in C++, so as to simulate the MAC layer communications. The radio wave propagation model utilized in the testbed is the *Log-distance Path Loss Model* as given in [15]. In our work all of the model's parameters are chosen to emulate a 2.4 GHz wireless radio, and if the distance is less than a certain threshold, the transmission power threshold is set to the minimum threshold of 1 dBm. The maximum transmission power is 29 dBm, which corresponds to a transmission distance of 1 mile.

V. Experimental Study

A. An Analysis of Clustering for 1-connectivity

For 1-connectivity, the centralized algorithm gives the lower bound of power consumption for both MinMax and MinTotal. Note that the centralized algorithm is analogous to treating the entire network as a single cluster. Table I shows the power increase percentages for MinMax generated by the five clustering algorithms in conjunction with *CLTC* over the centralized algorithm. The power increase percentages for MinTotal are very similar. Hence, our discussion in the remainder of this subsection will not distinguish between MinMax and MinTotal.

Among the five clustering algorithms, the performance of ADB is uniformly the best, with MMD finishing second followed by DDCA then WCA. In absolute terms, these four methods have performance that is much superior to clique. Note for example that with

Methods / # of Nodes	ADB	DDCA	WCA	clique	MMD
50	10%	16%	20%	48%	13%
100	16%	29%	37%	129%	26%
150	19%	44%	68%	222%	31%
200	23%	54%	83%	337%	54%

TABLE I
POWER INCREASE PERCENTAGE OVER THE CENTRALIZED ALGORITHM FOR MINMAX, 1-CONNECTIVITY

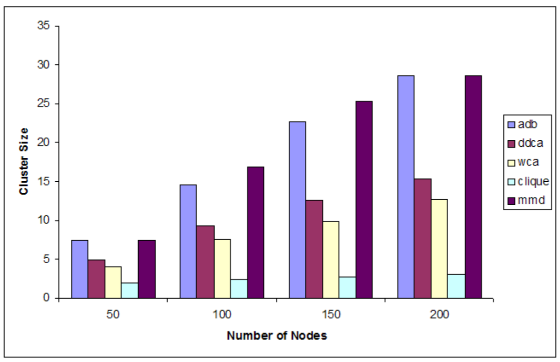

TABLE II
CLUSTER SIZE

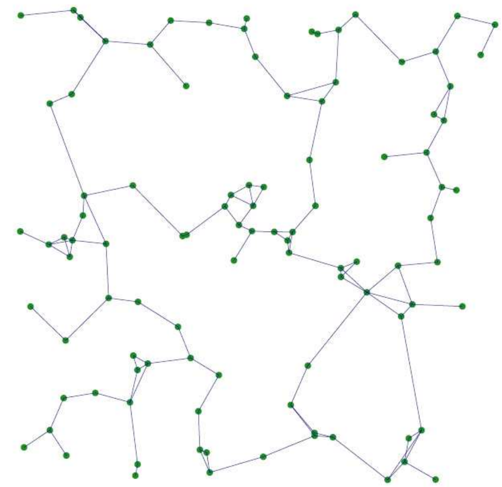

Fig. 1. The Network Resulting from ADB, MinTotal, 1-connectivity

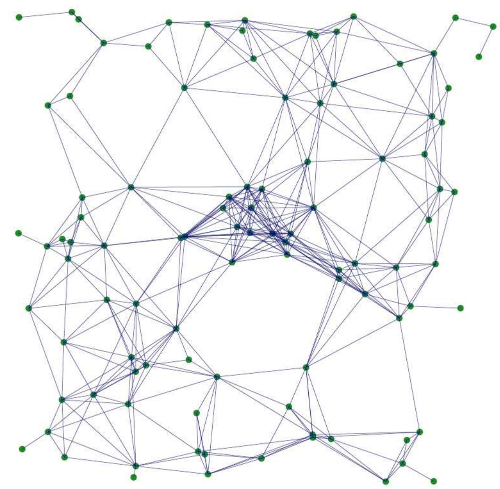

Fig. 2. The Network Resulting from clique, MinTotal, 1-connectivity

100 nodes, the maximum power assigned by clique is 129% greater than that of the centralized algorithm, while the other three methods range from 16% to 37%.

To gain insight into the effect of clustering on the performance of *CLTC*, average cluster sizes for the five methods are shown in Table II. Among the clustering methods, two hop clustering methods (i.e., ADB and MMD) generate larger clusters than one hop clustering methods and clique clustering. For example, for 200 node networks, the ADB clusters are 100% larger than DDCA clusters. Correspondingly, from Table I the power increase for DDCA over the centralized algorithm is slightly more than double the power increase for ADB over the centralized algorithm.

The primary conclusion from Tables I and II is that the *cluster size is the predominant factor influencing the performance*. Specifically, the larger the average cluster, the better the performance of *CLTC* in conjunction with that clustering method.

To see why the performance of clique clustering is poor relative to the other clustering methods, let us look at Figures 1 and 2. These two figures show the final network topologies as produced by ADB and clique clustering for MinTotal 1-connectivity. The average power of Figure 1 is 10.53, while the average power in Figure 2 is 21.31. From Table II the average cluster sizes generated by clique clustering are small, typically around three nodes for a cluster. In contrast, the average cluster size for the other three methods was nearly 17. Not surprisingly, the performance of clique clustering is much weaker than that of the other clustering methods.

We conclude that the best clustering algorithm for the *CLTC* framework must be based on a compromise between communication costs and cluster sizes. While larger cluster sizes lead to better performance, larger cluster sizes also require considerably more local information that needs to be exchanged and computed. In this context, it seems that two hop clusters are good enough, since for example, ADB generates sufficiently large cluster size that its performance is just slightly weaker than that of the centralized algorithm.

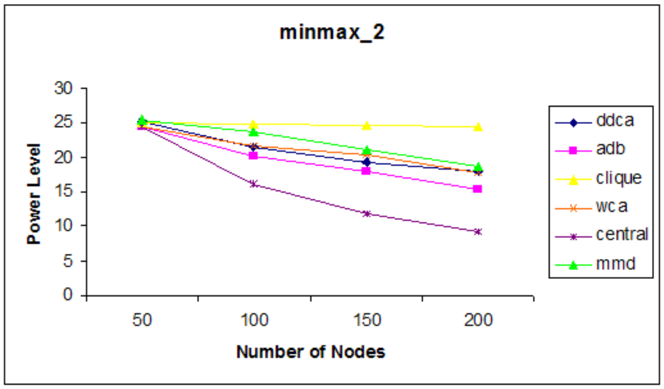

Fig. 3. MinMax 2-connectivity

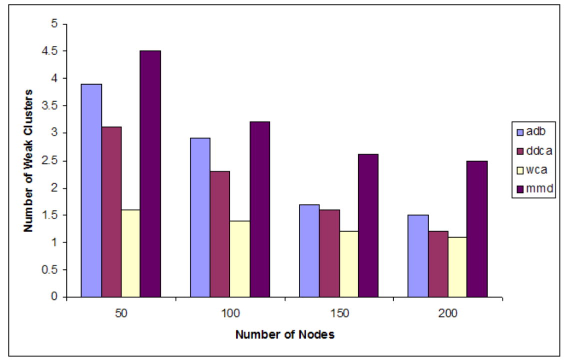

TABLE III

NUMBER OF WEAK CLUSTERS FOR MINMAX, 2-CONNECTIVITY

B. An Analysis of Clustering for 2-connectivity

This subsection discusses the impact of various clustering methods on the performance of *CLTC* when the goal is to produce a 2-connected network. These results will show that although cluster size continues to be the major factor in determining the performance, the node proximity of clusters also plays an important role for 2-connectivity, independent of cluster size.

We begin with Figure 3 which shows for 2-connectivity, the maximum power levels (MinMax) generated by the five clustering algorithms in conjunction with *CLTC* and the centralized algorithm.

1) An Analysis of MinMax for 2-connectivity: From Figure 3, for MinMax 2-connectivity, the performance of ADB is uniformly the best, with DDCA and WCA finishing second and third though neither is uniformly better than the other, followed by MMD. As with 1-connectivity, the performance of clique clustering is the worst. Note however that the performance gap between clique clustering and the other three clustering methods for 2-connectivity is not as severe as for 1-connectivity. Further, in general terms, there is a relationship between the performance and cluster size. However, that relationship is weaker than for 1-connectivity. Next, we explore the nature of this relationship.

We begin by comparing the performance of DDCA and WCA for MinMax 2-connectivity. Recall that DDCA generates slightly larger clusters than WCA (Table II). Hence, in general the performance of DDCA might be expected to be slightly stronger than that of WCA. Indeed, this was the case for 1-connectivity. However, for 2-connectivity there is an additional factor. Specifically, although DDCA generates larger clusters, hence fewer clusters, than WCA, it does generate more weak clusters than WCA (shown in Table III). Weak clusters reflect weak node proximity of clusters. This is a direct result of WCA generating geographically tighter clusters than DDCA. Recall that all of the nodes of a weak cluster, as well as neighboring nodes, are assigned full power. If the network has many weak clusters, then a large number of nodes will be assigned full power. Note that for 1-connectivity, there are no weak clusters, since there not being able to achieve the desired topology would mean that the cluster is not connected, which is an obvious contradiction to the very definition of a cluster. For 2-connectivity, both the cluster size and the node proximity of clusters play a role. When the network is less dense, DDCA generates considerably more weak clusters than does WCA (i.e. nearly twice as many for 50 node networks), and the performance of DDCA is weaker than that of WCA. Thus, although cluster size remains the primary factor determining performance for MinMax 2-connectivity, the node proximity of clusters also plays an important role.

2) An Analysis of MinTotal for 2-connectivity: Here, the experimental results show that no single clustering algorithm is uniformly the best. Nonetheless, it is clear that if the clustering can avoid producing weak clusters (i.e., maintaining strong node proximity), then it remains the case that the larger the cluster, the better the performance. Detailed analysis is omitted due to space constraints.

VI. DESIGN AND ANALYSIS OF MMST

In the prior section, we concluded that large cluster size and high node proximity of clusters are the most important determinants for a clustering method to work well in conjunction with *CLTC*. In this section, we introduce a new clustering method based on these two factors.

To achieve a sufficiently large cluster size while limiting the number of messages that need to be sent, we select 2-hop as the scope for collecting information about nearby nodes.

To achieve node proximity of clusters, we need to determine a way for selecting clusterheads that reflects this proximity. Recall that weight based clustering methods seem to provide better node proximity than do graph based methods (such as clique clustering

# of Nodes \ Methods	MinMax 2-connectivity		# of Nodes \ Methods	MinTotal 2-connectivity	
	ADB	MMST		WCA	MMST
50	0.6%	1.5%	50	-14%	-4%
100	25%	35%	100	-0.2%	14%
150	53%	42%	150	39%	23%
200	67%	62%	200	100%	79%

TABLE IV
POWER INCREASE PERCENTAGE OVER THE CENTRALIZED ALGORITHM FOR MINMAX AND MINTOTAL, 2-CONNECTIVITY

# of Nodes \ Methods	ADB	DDCA	WCA	clique	MMD	MMST
50	51%	47%	38%	15%	52%	52%
100	92%	85%	75%	20%	108%	109%
150	139%	112%	91%	22%	155%	121%
200	177%	159%	116%	24%	168%	163%

TABLE V
POWER INCREASE PERCENTAGE FOR MINMAX

and MMD). Nonetheless, node proximity "feels" like a graph based property. Our approach is to combine the weight based approach with the graph property approach in the following way: We let the weight of a node X be the *average edge length in a local minimum spanning tree of the 2-hop neighborhood of that node*. The intuition is that nodes with smaller such weights have neighborhoods that exhibit higher node proximity.

Formally, we define *A Multi-hop MST-based Clustering Algorithm for Ad Hoc Networks* (MMST), where each node performs the following procedure:

1) Collect information on all nodes in its 2 hop neighborhood.
2) Construct a local minimum spanning tree (MST) of the 2-hop neighborhood.
3) Compute its *node weight* as the average edge length in that local MST.
4) The node waits until all of the nodes of smaller weight in its 2-hop neighborhood have decided if they are a clusterhead. If there *is* at least one clusterhead among those nodes, then this node joins the cluster of the node of smallest weight. Otherwise, this node declares itself to be a clusterhead.

For 1-connectivity, MinMax and MinTotal, the performance of MMST is virtually identical to that of ADB. This is not a surprise, since both methods are based on 2-hop neighborhoods, and for 1-connectivity the impact of node proximity is relatively minor. For space reasons, we do not show the table with the data for 1-connectivity.

For 2-connectivity, the performance of MMST in relation to the clustering methods described in the prior sections is shown in Table IV. That table shows the performance of MMST in comparison with ADB for MinMax and with WCA for MinTotal. Recall that ADB is consistently the best of the earlier clustering methods for MinMax, and that WCA, ADB and DDCA are interchangeably the strongest for MinTotal. We use WCA as the representative of these three methods. The results in Table IV show that overall for 2-connectivity, MinMax and MinTotal, MMST is superior to the other methods in dense networks, though not for sparse networks. Note for example that for MinTotal, when the centralized algorithm does not beat WCA (i.e. negative percentages in Table IV), the performance of MMST is weaker than WCA. In contrast, when the centralized algorithm beats WCA, MMST is uniformly the best among all studied clustering methods[5]

VII. ENERGY AND INCREASED CONNECTIVITY

There is an obvious tradeoff between the desire that the network have the highest possible level of connectivity and the power used by the network nodes in achieving that level of connectivity. In this section we quantify this tradeoff, by examining the increase in power needed for a 2-connected network versus a 1-connected network.

For the various clustering methods studied in this paper, Table V shows the power increase percentage of 2-connectivity over 1-connectivity for MinMax, while the table for MinTotal is similar, hence omitted here. From these two tables it is clear that in general, a very high price is paid in requiring 2-connectivity instead of 1-connectivity.

For MinMax, the increases for the non-clique methods[6] range from 38-52% for sparse networks and steadily increase to be 116-177% for dense networks (200 nodes). It seems that for sparse networks, there are already a number of large power values even for 1-connectivity, hence there are already "extra" edges present in the 1-connected network that are there as an artifact of these large power values. These edges are useful in providing 2-connectivity, hence reducing the need for additional edges, hence limiting the needed increase in power values. By contrast for dense networks, the power values to achieve 1-connectivity are small, and there are few "extra" edges, hence in providing 2-connectivity, most the edges must be

[5]Note that: WCA gives the best performance for dense networks.

[6]The only exception to the dramatic increase is for clique clustering. Recall that clique clustering for 1-connectivity produces many edges. Subsequently, the number of additional edges needed to establish 2-connectivity is small, and yet the total number of edges remains large relative to the other methods.

produced by the increased power values. As a consequence, these power values are proportionally larger. Indeed for the best methods for MinMax, namely ADB and MMST, the increases for dense networks are 177% and 163%, respectively. Obviously, these increases in power are quite large indicating that requiring a 2-connected topology should be done only when there is a compelling reason.

For MinTotal, the story is similar, as the increases for non-clique methods range from 39-73% for sparse networks and from 136-253% for dense networks. Details are omitted due to space constraints.

VIII. Related Work

Earlier work on topology control has taken two contrasting approaches. One group of works has studied *centralized* algorithms where a single source has complete information about the entire network. That source computes power assignments which are then transmitted to the other nodes. These works have typically focused on proving specific bounds about the goodness of their solutions relative to optimal. The other approach has been to give *fully distributed* algorithms where nodes have only local knowledge of the network. Here some heuristic calculation is used to determine the power assignment at each node, and the goodness of the assignment is established using simulations.

A. Centralized algorithms

In the earliest formal work in this area, Ramanathan and Rosales-Hain [17] formulated the topology control problem for MinMax 1-connectivity and 2-connectivity. They developed two optimal centralized algorithms. In [21], Wieselthier *et al.* introduced and evaluated algorithms for broadcast and multicast trees in wireless networks for MinTotal but no performance bound was guaranteed. Additional work for centralized topology control may be found in [13], [6], [5], [9].

B. Fully distributed heuristics

Fully distributed heuristics for topology control can be divided into two categories: those using only link costs in their decision making (called *cost-based heuristics*) [16], [12], and those using node location information and properties of two dimensional geometric graphs (called *location-based heuristics*) [20], [19]. For both types of methods, the goal is to allow mobility or at least dynamic joining and leaving of nodes. It is usually assumed that the network is always connected and that each node collects 1-hop information through periodic, asynchronous "Hello" messages.

Disclaimer: The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government.

References

[1] A.D. Amis, R. Prakash, T.H.P. Vuong and D.T. Huynh. "Max-Min D-Cluster Formation in Wireless Ad Hoc Networks", *Proc. of INFOCOM*, March 2000, pp. 32–41.

[2] S. Basagni. "Distributed clustering for ad hoc networks", Proc. of IS-PAN, Perth/Fremantle, Australia, June 1999.

[3] M. Chatterjee and S. Das and D. Turgut. "WCA: A Weighted Clustering Algorithm for Mobile Ad hoc Networks", *Journal of Cluster Computing, Special issue on Mobile Ad hoc Networking*, Vol. 5, No. 2, April 2002, pp. 193–204.

[4] Y. P. Chen and A. L. Liestman. "Approximating minimum size weakly-connected dominating sets for clustering mobile ad hoc networks", MobiHoc'02, June 2002, pp. 165–172.

[5] A. E. F. Clementi, P. Penna and R. Silvestri. "Hardness Results for the Power Range Assignment Problem in Packet Radio Networks", *Proc. Third International Workshop on Randomization and Approximation in Computer Science* (APPROX 1999), Lecture Notes in Computer Science Vol. 1671, Springer-Verlag, July 1999, pp. 195–208.

[6] G. Calinescu and P-J. Wan. "Range Assignment for High Connectivity in Wireless Ad Hoc Networks", *Adhoc-Now '03*.

[7] Y. Fernandess and D. Malkhi. "K-Clustering in Wireless Ad Hoc Networks", Proc. ACM IW-POMC, 2002, pp. 31–37.

[8] C. Jaikaeo and C.-C. Shen. "Adaptive Backbone-based Multicast for Ad hoc Networks", *Proc. IEEE International Conference on Communications (ICC 2002)*, April 2002, 7 pages.

[9] S. O. Krumke, R. Liu, E. L. Lloyd, M. V. Marathe, R. Ramanathan and S. S. Ravi, "Topology Control Problems Under Symmetric and Asymmetric Power Thresholds", *Proc. International Conference on Ad hoc and Wireless Networks* (ADHOC-NOW'03), Lecture Notes in CS, Vol. 2865, (Edited by S. Pierre, M. Barbeau and E. Kranakis), Montreal, Canada, Oct. 2003, pp. 187–198.

[10] R. Krishna and N. Vaidya and M. Chaterjee and D. Pradhan. "A Cluster-based Approach for Routing in Dynamic Networks" *ACM SIGCOMM Computer Communication Review*, April 1997, pp. 49–65.

[11] C. H. R. Lin and M. Gerla. "A Distributed Control Scheme in Multi-hop Packet Radio Networks for Voice/Data Traffic Support", IEEE GLOBECOM, pp. 1238-1242, 1995.

[12] N. Li, J. C. Hou and L. Sha. "Design and Analysis of An MST-Based Topology Control Algorithm", *Proc. of INFOCOM*, March/April 2003, Vol. 3, pp. 1702–1712.

[13] E. L. Lloyd, R. Liu, M. V. Marathe, R. Ramanathan and S. S. Ravi, "Algorithmic Aspects of Topology Control Problems for Ad Hoc Networks", *Mobile Networks and Applications* (MONET), Vol. 10, Issue 1-2, Feb.–Apr. 2005, pp. 19–34.

[14] A. B. McDonald and T. F. Znati. "Design and Performance of a Distributed Dynamic Clustering Algorithm for Ad-Hoc Networks", Proc. 34th Annual Simulation Symp., Apr. 2001, pp. 27–35.

[15] T. S. Rappaport. *Wireless Communications: Principles and Practice*, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1996.

[16] V. Radoplu and T. H. Meng. "Minimum Energy Mobile Wireless Networks", *IEEE J. Selected Areas in Communications*, 17(8), Aug. 1999, pp. 1333-1344.

[17] R. Ramanathan and R. Rosales-Hain. "Topology Control of Multihop Wireless Networks Using Transmit Power Adjustment", *Proc. IEEE INFOCOM 2000*, Tel Aviv, Israel, March 2000, pp. 404–413.

[18] C.-C. Shen and C. Srisathatpornphat and R. Liu and Z. Huang and C. Jaikaeo and E. Lloyd. "CLTC: A Cluster-based Topology Control Framework for Ad hoc Networks", *IEEE Trans. on Mobile Computing* 3(1) 2004, pp. 18–32.

[19] R. Wattenhofer, L. Li, P. Bahl and Y. Wang. "Distributed Topology Control for Power Efficient Operation in Multihop Wireless Ad Hoc Networks", *Proc. IEEE INFOCOM 2001*, Anchorage, Alaska, April 2001, pp. 1388–1397.

[20] Y. Wang, X. Li, P. Wan and O. Frider. "Distributed Spanners with Bounded Degree for Wireless Ad Hoc Networks", *International Journal of Foundations of Computer Science*, Vol. 14, No. 2, pp. 183–200, 2003.

[21] J. E. Wieselthier, G. D. Nguyen and A. Ephremides. "On The Construction of Energy-Efficient Broadcast and Multicast Trees in Wireless Networks", *Proc. of INFOCOM*, March 2000, pp. 585–594.