A New Performance Metric for Construction of Robust and Efficient Wireless Backbone Network

Ziyi Zhang, *Member, IEEE,* Xin Wang, *Member, IEEE,* and Qin Xin, *Member, IEEE*

***Abstract*— With the popularity of wireless devices and the increasing demand of network applications, it is emergent to develop more effective communications paradigm to enable new and powerful pervasive applications, and to allow services to be accessed anywhere, at anytime. However, it is extremely challenging to construct efficient and reliable networks to connect wireless devices due to the increasing communications need and the dynamic nature of wireless communications. In order to improve transmission throughput, many efforts have been made in recent years to reduce traffic and hence transmission collisions by constructing backbone networks with the minimum size. However, many other important issues need to be considered. Instead of simply minimizing the number of backbone nodes or supporting some isolated network features, in this work, we exploit the use of algebraic connectivity to control backbone network topology design for concurrent improvement of backbone network robustness, capacity, stability and routing efficiency. In order to capture other network features, we provide a general cost function and introduce a new metric, connectivity efficiency, to tradeoff algebraic connectivity and cost for backbone construction. We formally prove the problem of formulating a backbone network with the maximum connectivity efficiency is NP-hard, and design both centralized and distributed algorithms to build more robust and efficient backbone infrastructure to better support the application needs. We have made extensive simulations to evaluate the performance of our work. Compared to literature studies on constructing wireless backbone networks, the incorporation of algebraic connectivity into the network performance metric could achieve much higher throughput and delivery ratio, and much lower end-to-end delay and routing distances under all test scenarios. We hope our work could stimulate more future research in designing more reliable and efficient networks.**

Our performance studies demonstrate that, compared to peer work, the incorporation of algebraic connectivity into network performance metric could achieve much higher throughput and delivery ratio, and much lower end-to-end delay and routing distances under all test scenarios. We hope our work could stimulate more future research in designing more reliable and efficient networks.

I. INTRODUCTION

THere are increasing interests and use of wireless networks with the proliferation of wireless devices, and the fast progress of mobile computing and wireless networking techniques. In a multi-hop wireless network, wireless devices could self-configure and form a network with an arbitrary topology. The network's topology may change rapidly and unpredictably. Such a network may operate in a stand-alone fashion, or may be connected to the larger Internet. Multi-hop wireless networks become a popular subject for research in recent years, and various studies have been made to increase the performance of the networks and support more advanced mobile computing and applications [1]–[5].

With the popularity of wireless devices, there is a significant increase of communication nodes in the network. In wired networks, a larger number of network nodes could potentially lead to the increase of throughput and reduction of network diameter. In wireless networks, however, due to the sharing of transmission medium, the competition among a larger number of nodes in accessing the channel would result in higher transmission collisions, thus a significant increase of transmission delay, throughput degradation, and extra energy consumption. Many efforts have been made in recent years to construct a backbone network to carry the total network traffic by selecting a minimum set of backbone nodes out of the total network nodes, in order to reduce the total network transmissions and hence collisions for improving the network throughput [**?**], [10]–[13], [26]–[28].

In a dynamic wireless network, it is important and challenging to support reliable communications to reduce network transmission loss and failure in presence of node mobility, device unreliability and unstable wireless communications medium. It is especially important to ensure reliable communications over a backbone network which is responsible for carrying the total network traffic. In addition, there is a need to consider other important factors in backbone design, including network stability, capacity, load balancing, path length, energy consumption and therefore longevity. These issues are largely ignored in topology studies carried in the literature. Although a very limited number of backbone schemes [17], [24], [37] have been proposed to support a certain degree of network reliability, the increase of vertex degree of backbone nodes locally is conservative, which would compromise the network capacity with a higher number of backbone nodes. They also generally do not consider other important network features discussed above.

To achieve backbone reliability and support other important network features discussed above, in this work, we propose a new metric, *connectivity efficiency*. Specifically, we exploit use and control of algebraic connectivity [15], an important concept introduced in spectral graph theory,

Ziyi Zhang is currently with Stony Brook University, Stony Brook, New York, USA

Xin Wang is currently with Stony Brook University, Stony Brook, New York, USA

Qin Xin is currently with Université Catholique de Louvain, Louvain-la-Neuve, Belgium.

in backbone network design to control backbone network topology for improved robustness, stability, capacity and routing efficiency. To further capture other network features, we consider the use of a cost function. The connectivity efficiency is defined as a function of algebraic connectivity and total network cost, which allows the backbone design to tradeoff between increasing algebraic connectivity and reducing total network cost. To the best of our knowledge, this is the first work that exploits use of algebraic connectivity to capture the spectral characteristics of the network graph in designing a wireless backbone network that can simultaneously improve the network performance from several important perspectives. In addition, the introduction of a general cost function allows the incorporation of other network features in backbone design. Besides serving as a basis for designing an efficient and reliable mobile backbone network, we expect the new performance metric proposed in this work to be applicable in other network research fields for a higher network performance.

The objective of our backbone design is to select a subset of nodes out of all candidate backbone nodes to form a connected backbone network with the maximum network connectivity efficiency. We prove that the connectivity efficiency maximization problem is NP-hard, and propose both centralized and distributed approximate algorithms to solve the problem. Our backbone construction algorithms do not constrain the cost function format so that the proposed backbone formulation algorithms can be applied for achieving various design goals. To demonstrate the benefit of introducing a new and more effective metric for backbone design and evaluate the performance of our backbone construction algorithm, we introduce a node cost model to capture the impact on delay and hence network capacity due to node capacity, transmission error, and node distribution. The total network cost is the summation of node cost.

The rest of the paper is organized as follows. In Section II, III, and IV, we review the related work in the literature, analyze the features of algebraic connectivity and formulate the problem. We prove that the NP hardness of the connectivity efficiency maximization problem in Section V, and present our centralized algorithm and distributed algorithm in Sections VI and VII respectively. In Section VIII, we evaluate the performance of our algorithms through extensive simulations. Finally, we summarize the results and discuss future research directions in Section IX.

II. RELATED WORK

Cluster organization has been widely studied in the literature work. It is generally performed in two steps, selecting cluster heads among nodes based on some criteria and forming clusters by associating each cluster head with a set of members. Clusterhead selection criteria fall into three categories: lowest (or highest) ID among all unassigned nodes [30], maximum node degree [29], or some generic weight [25]. A set of heuristic approaches have been proposed to construct the backbone networks. Ju et al. [7], [39] introduced heuristic approaches to construct the backbone network. Blough et al. [18] intended to constrain the interference by limiting the maximum degree of nodes.

Distributed algorithms to construct connected dominating sets (CDS) in ad hoc networks are studied in [10]–[12]. Alzoubi et al. [10] models the transmission range as a unit disk, and proposes a localized approximate method to construct the minimum CDS within a constant time using a linear number of messages. Marathe et al. [34] also models the network as unit disk graph, and considers methods for constructing maximum independent set, minimum coloring, and minimum dominating set. The algorithm in [11] marks a node as a dominator if it has two unconnected neighbors, and reduces the CDS size by applying two dominant pruning rules. Dai et al. [12] further improve the algorithms proposed in [11] to reduce CDS size. Wu et al. [1] propose an iterative local solution (ILS) for computing a CDS with the objective of reducing the CDS size over a number of iterations. A survey and simulation-based performance studies were carried in [31] to compare various backbone construction schemes proposed in literature. Scheideler et al. [19] further explored interference model in the dominating set problem. These schemes mainly focus on forming a CDS with the minimum size without considering the transmission reliability and other network features.

Algorithms in [13], [25]–[28] considered using different weights as the priority criteria to select clusterheads, while the goal of the majority of the schemes is to minimize the number of clusterheads (or the size of the backbone) instead of the total weight of the clusterheads. The priority is given to nodes with high stability or low mobility in [27], and to nodes relatively stable and with high degree in [28]. Basagni [26] gives an algorithm to solve the maximal weighted independent set problem. Wang et al [13] develops a distributed heuristic algorithm for constructing the minimum weighted dominating set and the minimum weighted connected dominating set. However, these algorithms also do not consider the overall backbone network reliability.

The authors in [36], [37] observe the importance of reducing network diameter. The backbone construction in [36] is based on the hard limit on the network diameter without considering reliability. The authors in [17], [24] and [37] intended to form a more robust backbone network which was k-connected, k-dominating and k-connected, m-dominating respectively by enforcing a conservative local vertex degree constraint. In contrast, we exploit the algebraic connectivity which supports transmission reliability at a larger network range (i.e., the path level) in our backbone design to improve network reliability. The algebraic connectivity has a continuous value and can serve as a fine metric to measure the network connectivity. Besides serving a metric for network reliability, a higher algebraic connectivity will ensure a higher bottleneck capacity and network capacity, higher network stability, and higher routing efficiency. The reduction of network diameter is a natural result when increasing the algebraic connectivity for a higher network reliability. In addition to these factors, our proposed performance metric incorporates a general cost function to allow the capture of other network features in the backbone design with a tradeoff between increasing algebraic connectivity and reducing the total network cost.

The idea of constructing a hierarchical backbone network was considered in [32], [33], [35], [38]. Xu et al. [38] simply selects the nodes that first claim the leadership in a neighborhood to be clusterheads, while TBONE proposed in [35] attempts to minimize the number of backbone nodes, giving priority to higher weight nodes. Work in [32] attempts to cover all the regular nodes assuming there are an infinite number of backbone capable nodes, while minimizing the number of nodes required in the backbone construction. The authors in [33] propose a more practical backbone network deployment algorithm with a given number of bachbone nodes. Instead of simply minimizing the number of backbone nodes or some basic cost function, we propose a new performance metric in this paper, which will facilitate building more reliable and efficient backbone networks. Finally, the problem of finding the maximum algebraic connectivity augmentation has been proved to be NP-hard in [45]. In this paper, we further prove that forming a connected dominating set by selecting a subset of nodes from the network to maximize the algebraic connectivity efficiency is NP-hard.

III. Theoretical Foundation

In this section, we analyze the properties of algebraic connectivity which are important for network design. We first introduce some basic concepts in graph theory, and we then show the good features of algebraic connectivity and its properness in being a backbone network design metric through the analysis of a set of theories on the characteristics of algebraic connectivity.

For a graph G with n vertices, $v(G)$ and $e(G)$ are vertex and edge connectivity of G respectively. The diameter $diam(G)$ equals the maximum shortest distance between all pairs of vertices, and $\overline{\rho}$ represents the average distance. Spectral graph theory studies the properties of a graph G in relationship to the characteristic polynomial, eigenvalues, and eigenvectors of its adjacency matrix A or Laplacian matrix L. The Laplacian of G is defined as $L(G) = \Delta - A$, where the elements of the diagonal matrix Δ are the vertex degrees of G with Δ_m as the maximum of them, and L is positive semidefinite quadratic. Assume L has n eigenvalues ordered with multiplicity, $\lambda_1 \leq \lambda_2 \cdots \leq \lambda_{n-1} \leq \lambda_n$, in [15], Fiedler coined algebraic connectivity as $a(G) = \lambda_2$ which is a non-negative real number.

To justify that $a(G)$ is a good measure of graph connectivity, Fiedler and Weyl [8], [15] provided several properties as follows.

Lemma 1: If $G_1 = (V, E_1)$ and $G_2 = (V, E_2)$, $a(G_1) + a(G_2) \leq a(G_1 \cup G_2)$.

Theorem 1: For $G_1 = (V, E_1)$ and $G_2 = (V, E)$, if $E_1 \subset E$, $a(G_1) \leq a(G_2)$.

Theorem 2: (Interlacing Theorem) If $G' = G + e$, $\lambda_i(G) \leq \lambda_i(G') \leq \lambda_{i+1}(G)$, $i = 1, \ldots, n-1$.

For a network, as the number of connections increases, the level of connectivity should not decrease. This property has been exploited in [44] to increase the algebraic connectivity of the network. As the algebraic connectivity $a(G)$ does not drop when the edge set E becomes larger, it is a good metric to capture network connectivity. Normally, the addition of an extra connection will not significantly change the network connectivity level unless a critical edge that can merge two disconnected components of the network is added. Based on the algebraic connectivity theory, the number of times 0 appears as an eigenvalue in the Laplacian represents the number of connected components in the graph. The addition of one critical edge will reduce the number of connected components by one, and a connected network has only one connected component. The algebraic connectivity will change from 0 to be larger than 0, once the network becomes connected. The interlacing theorem ensures that $\lambda_2(G')$ is bounded between $\lambda_2(G)$ and $\lambda_3(G)$, which indicates that algebraic connectivity is not too sensitive to a small change to the network, which can be frequent in a dynamic network, unless it is critical.

For a network to be reliable, it is desirable to have a higher edge and/or vertex connectivity in order to handle link or node failure. This is particularly important for mobile wireless networks. In [15], Fiedler provided the following theory to provide the bounds and relate $a(G)$ to the conventional connectivity measures $v(G)$ and $e(G)$:

Theorem 3: The following conditions hold.

(1) $a(G) \leq v(G) \leq e(G)$

(2) $a(G) \geq 2e(G)(1 - cos\frac{\pi}{n})$

(3) $a(G) \geq 2(cos\frac{\pi}{n} - cos\frac{2\pi}{n})e(G) - 2cos\frac{\pi}{n}(1 - cos\frac{\pi}{n})\Delta_m$.

The following theorems proposed by Kirchhoff [9], Alon and Milman [21] correlate the structure of the graph with algebraic connectivity.

Theorem 4: (Matrix Tree Theorem) The number of spanning trees $t(G) = \frac{1}{n}\Pi_{i=2}^{n}\lambda_i$.

Theorem 5: If $G = (V, E)$, $A, B \subset V$, $A \cap B = \phi$, F represents the set of edges that do not have both ends in A or B, then $|F| \geq \rho^2 \lambda_2 \frac{|A||B|}{|A|+|B|}$, where ρ is the minimum distance between A and B.

Theorem 6: $|\partial A| \geq \lambda_2 \frac{|A|(n-|A|)}{n}$, where ∂A is the edge cut induced by A and $V - A$.

The number of spanning trees represents the number of ways to connect a pair of vertices in the graph. For a network to be reliable, it is desirable to have multiple paths between nodes in order to establish an alternative path upon route breakage or congestion. Since $a(G)$ is the smallest multiplier in Theorem 4, $\frac{a(G)^{n-1}}{n}$ serves as a lower bound of $t(G)$. In Theorems 5 and 6, there are more edges in the edge cut if $a(G)$ is larger, which implies that a network with a larger algebraic connectivity is not likely to be partitioned. In addition, a larger cut would lead to a higher flow capacity according to the max-flow min-cut theorem. That is, in a flow network, the maximum amount of flow passing from the source to the sink is equal to the minimum capacity that needs to be removed from the network so that no flow can pass from the source to the sink. It has been shown that the network capacity heavily depends on the min-cut set in [22].

Some recent discoveries by Mohar [14] indicate that $a(G)$ has close relationship with routing problem.

Theorem 7: $diam(G) \leq 2\lceil\sqrt{\frac{\lambda_n}{\lambda_2}\frac{\alpha^2-1}{4\alpha}}+1\rceil\lceil log_\alpha \frac{n}{2}\rceil$, where $\alpha > 1$.

Theorem 8: $diam(G) \leq 2\lceil\frac{\Delta+\lambda_2}{4\lambda_2}ln(n-1)\rceil$.

Theorem 9: $\overline{\rho} \leq \frac{n}{n-1}\lceil\frac{\Delta+\lambda_2}{4\lambda_2}ln(n-1)\rceil$.

These theorems provide the upper bound for the graph diameter and average distance, and the upper bound reduces as the algebraic connectivity increases. This property is very important for network design as it is highly desirable to bound the distance or the number of hops between two network nodes.

In addition to serving as an index for network reliability, algebraic connectivity can also reflect network stability and robustness, as the effect of the dynamics of a node is averaged out rapidly and thus has a minor influence on the stability of a network with a large algebraic connectivity [20].

In summary, algebraic connectivity is a good metric for measuring the network performance. Compared to conventional connectivity measures such as vertex connectivity and edge connectivity, it has a continuous value and provides a fine metric to measure the network connectivity level. It not only captures the network connectivity, but also to some extent, reflects the network stability and gives a lower bound on the performance of the network bottlenecks. The latter impacts the overall network capacity. Additionally, algebraic connectivity controls the upper bound of the network routing distance. Therefore, algebraic connectivity can capture some important features of the network, including robustness, capacity, and routing efficiency. As a result, algebraic connectivity can serve as a good design metric for mobile wireless networks, and the network performance can be improved by constructing a network with a larger algebraic connectivity. We demonstrate through our performance studies in Section VIII that algebraic connectivity can help to effectively improve the network reliability while not significantly reducing the capacity, and could also help to reduce the routing distance.

IV. PROBLEM FORMULATION

In light of above discussions, the objective of our work is to exploit the use of algebraic connectivity in backbone design to improve backbone network robustness, capacity, and routing efficiency. Additionally, we incorporate a cost function into the design metric to capture some other desired network features. The backbone design will compromise between increasing network algebraic connectivity by including more nodes into the backbone and reducing total network cost by removing nodes that incur high cost. As different backbone features would be needed by different applications, to make our algorithm general, we will not constrain the format of cost functions but will use a general cost function $C(\cdot)$ during our algorithm introduction.

Based on node capabilities, we divide wireless nodes in the network into two types. The first type of nodes is called backbone capable nodes (BCNs), which generally have higher capacity and can transmit at longer range. The second type of nodes is called regular nodes (RNs), which normally have lower capacity and transmit at shorter range. In a pervasive computing environment, the regular nodes can be simple sensors, or low power wireless devices, while the backbone capable nodes can be devices with more energy such as devices plugged in the outlets of offices or cars, with more capacity such as laptops and wireless gateways, and/or transmitting at larger range such as 802.11 nodes (as compared to 802.15.4-based sensor nodes) and WiMAX nodes. We focus on the construction of more robust and efficient backbone network by properly selecting backbone nodes among BCNs. As this work focuses on backbone construction, we assume all the nodes are backbone capable nodes, i.e., BCNs. For the convenience of presentation, we do not specially identify BCNs in the remaining of the paper.

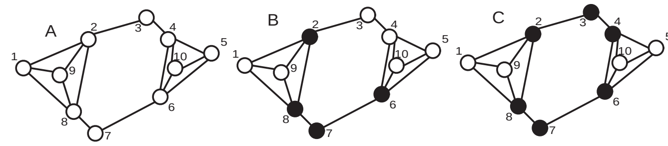

Fig. 1. Example Backbone Construction

We first introduce some concepts and terminologies to be used in the remaining of the paper. For a graph $G = (V, E)$, define the cost of a node i as c_i for $\forall v_i \in V$, $i = 1, 2, \cdots |V|$, and the total cost of the graph G as $C(G) = \sum_{v_i \in V} c_i$. Based on the analysis of Section III, to improve the robustness, capacity, and routing efficiency of a graph G while reducing its total cost, we define a new metric called *connectivity efficiency* (CE) as

$$\gamma(G) = \frac{a(G)}{C(G)}. \quad (1)$$

A subset D of the vertices in graph G is a dominating set (DS) if each node in the graph is either an element of D or is adjacent to some element of D. Dominators are elements in the set D and dominatees are not in it. A connected dominating set (CDS) is a dominating set whose elements induce a connected graph.

The backbone network construction problem considered in this work is to find a subset of network nodes that can form a connected dominated set with the objective of maximizing the connectivity efficiency. We call the problem *efficient connected dominated set* building problem, or ECDS. Our backbone construction problem can be formally presented as follows.

Problem Statement 1: ECDS: For a graph $G = (V, E)$, find a sub-graph $G^* = (D, E^*)$ induced by dominating set D that maximizes $\gamma(G^*)$.

TABLE I
EXAMPLE NODE COST.

Node	1	2	3	4	5	6	7	8	9	10
Cost	2	2	4	2	4	2	3	3	7	2

Before presenting the details of the problem, we will show the significance of our problem through an example. A backbone needs to be constructed for the example network in Fig.1A, with the cost of each node randomly set as in table IV. A backbone is considered functional if it covers all the nodes in the network and is connected, and not fully functional if it does not meet either of the requirements. If a backbone network does not meet the second requirement, it is considered disconnected.

For backbone construction, the low-cost algorithm (e.g. [13]) will, for example, select {2, 6, 7, 8} as backbone nodes (marked dark in figure) with the minimum cost of 10. If any of the four nodes is down, the network will not be completely covered. If node 7 or 8 is down, the backbone will be disconnected. Therefore, the backbone constructed by only minimizing the cost is vulnerable to failure.

In ECDS, reliability is one of the important consideration factors and the backbone set selected (Fig. 1C) is {2, 3, 4, 6, 7, 8} with the highest connectivity efficiency of 0.0625. A failure of any of the six nodes will not impact the functionality of the backbone. Considering all possible scenarios of two-node failure, with the probability of 0.27 and 0.4, simultaneous failures from two nodes will not impact the function and connection of the backbone network respectively. This indicates that a backbone network constructed using ECDS can tolerate better to the node failure.

From Fig. 1B, we also observe that the worst routing path between two nodes, e.g. the one between node 3 and node 5, has 5 hops while the shortest path between these two nodes has only 2 hops. The average routing distance between all pairs of vertices is 2.38 hops, with a 0.31 hop increase from that of the original topology. While in ECDS case, the longest routing distance is 4 hops, which is equal to the diameter of the original network. The average routing distances is 2.07 which is the same as that of the original one. The low cost algorithm also has several critical edges and nodes. In ECDS case, the minimum cut has two edges or two nodes. This example demonstrates that it is important to construct a more reliable backbone network with higher bottleneck capacity and routing efficiency. ECDS is designed to facilitate the construction of a backbone network with the desired features.

As mentioned earlier, our backbone construction algorithm is not constrained by a specific cost function. For evaluating the efficiency of our backbone construction algorithm, in this work, we choose node delay as cost and consider a node cost model that incorporates the following factors in order to balance network traffic and reduce transmission delay.

Node Capacity. We define a *Transmission Delay Factor* of a node i, (f_t^i) as $f_t^i = \frac{1}{W^i} = \frac{1}{\sum_{ij \in E} W_{ij}}$, with W_{ij} being the link transmission rate between node i and its neighbor node j. The higher the transmission rate on a node, the lower the delay.

Retransmission. Retransmission due to packet loss and error increases the delay of a packet. The packet loss is impacted by network load. With the loss and error rate p_e^i of a link measured, the expected number of transmissions can be calculated as $\frac{1}{1-p_e^i}$, and used as the retransmission delay factor f_r^i.

Node Distribution. When nodes share the transmission medium, the competition among nodes leads to extra delay. Assuming in a neighborhood there are N_c active nodes which have packets to send and share the same channel, if each node i is given a transmission weight w_i for a relatively long period, the transmission opportunity for node i can be represented as: $p_c^i = \frac{w_i}{\sum_{k=1}^{N_c} w_k}$. If CSMA based scheme is used, the delay factor (f_c) due to node distribution and competition can be estimated as $f_c = \frac{1}{p_c^i}$, which can be estimated based on the network topology and traffic.

By combining all major delay factors mentioned above, the cost of a node is defined as

$$W_i = f_t^i \cdot f_r^i \cdot f_c^i = \frac{1}{W^i} \cdot \frac{1}{1-p_e^i} \cdot \frac{1}{p_c^i} \qquad (2)$$

Generally, reducing the transmission delay would help improve network throughput. In a wireless network, a higher number of nodes in a neighborhood could potentially increase the collision, and reduce the network throughput. On the other hand, increasing algebraic connectivity helps to improve bottleneck throughput and reduce the path length, which will help improve network throughput. With the use of both algebraic connectivity and the above cost model in the backbone metric, our backbone construction algorithm intends to build a more reliable backbone network while achieving a higher network throughput and routing efficiency.

V. Problem Hardness

The objective of ECDS is to find a connected dominating set of a network graph that has the maximum connectivity efficiency. The search of the optimal solution only involves the selection of vertices (i.e., nodes), not any edge. In the following, we will first prove that ECDS is NP-hard, and we will then propose a centralized and a distributed approximation algorithms in Section VI and VII to build the backbone network, by selecting an appropriate set of nodes from the network to form the CDS with maximum connectivity efficiency.

We use K_n to represent a complete graph with n vertices. A clique is a complete sub-graph of G, and the Maximum Clique (MC) problem is to find in G a clique with the maximum number of vertices. We begin with the introduction of several lemmas. We will show that a special instance of ECDS problem is a maximum clique (MC) problem, which is known to be NP-hard.

Lemma 2: For a graph $G = (V, E), |V| = n$, if $G \neq K_n$, then the algebraic connectivity of the graph $a(G) < n$. **Proof:** For a graph $G \neq K_n$, we have $v(G) < n$. The relationship $a(G) < n$ directly follows due to Theorem 3, where $a(G) \leq v(G)$. □

Lemma 3: For a graph $G = (V, E), |V| = n \geq 3$, assume the cost value associated with node k is $c_k = 1$, and for a node $v_i \in V$ and $i \neq k$, the cost is $c_i = \epsilon < 1$

with $\epsilon \rightsquigarrow 1$. Then $\gamma(G) < \frac{n}{(1-\epsilon)+n\epsilon}$ if $G \neq K_n$, and $\gamma(G) = \frac{n}{(1-\epsilon)+n\epsilon} \rightsquigarrow \frac{1}{\epsilon}$ if $G = K_n$.

Proof: According to Lemma 2, if $G = (V, E), |V| = n$ is not a complete graph, $a(G) < n$. As $C(G) = \sum_i c_i = (1-\epsilon) + n\epsilon$, so $\gamma(G) = \frac{a(G)}{C(G)} < \frac{n}{(1-\epsilon)+n\epsilon}$. If $G = K_n$, i.e., G is a complete graph, $a(G) = n$, so $\gamma(G) = \frac{n}{(1-\epsilon)+n\epsilon} \rightsquigarrow \frac{1}{\epsilon}$, with $\epsilon \rightsquigarrow 1$. This shows that a complete graph will maximize the connectivity efficiency of a uniform-cost graph.

In what follows, we show that the maximum clique under a special instance is an induced solution of ECDS. This observation will be used to construct a special example to show MC problem is Turing reducible to ECDS in Theorem 10.

Lemma 4: For a graph $G = (V, E)$, assume the cost value associated with node k is $c_k = 1$, and for any other nodes $v_i \in V$ and $i \neq k$, the cost is $c_i = \epsilon < 1$ with $\epsilon \rightsquigarrow 1$. **If** node v_k must be selected to construct the dominating set and the maximum clique denoted by Q_M^k containing v_k is a dominating set of V, then the CDS containing v_k and constructed based on ECDS will induce Q_M^k.

Proof: As Q_M^k is assumed to be the maximum clique containing v_k and a dominating set of V in the special instance we construct here, according to Lemma 3, its connectivity efficiency is $\gamma(Q_M^k) = \frac{|Q_M^k|}{(1-\epsilon)+|Q_M^k|\epsilon} \rightsquigarrow \frac{1}{\epsilon}$, where $|Q_M^k|$ is the cardinality of its vertex set. Note that, in general, the maximum clique may not be a dominating set. Assume another sub-graph containing v_k and denoted by S^k is the induced graph of ECDS, i.e., the connected dominating set that has the maximum γ. If $S^k = K_{|S^k|}$, i.e., S^k is a complete graph, then $\gamma(S^k) = \frac{|S^k|}{(1-\epsilon)+|S^k|\epsilon} < \frac{|Q_M^k|}{(1-\epsilon)+|Q_M^k|\epsilon} = \gamma(Q_M^k)$, as $|S^k| < |Q_M^k|$ due to the maximum clique Q_M^k and the function $\frac{x}{(1-\epsilon)+x\epsilon}$ is monotonically increasing. If $S^k \neq K_{|S^k|}$, based on Lemma 3, $\gamma(S^k) < \frac{|S^k|}{(1-\epsilon)+|S^k|\epsilon} < \frac{1}{\epsilon}$ while $\gamma(Q_M^k) = \frac{|Q_M^k|}{(1-\epsilon)+|Q_M^k|\epsilon} \rightsquigarrow \frac{1}{\epsilon}$ for large enough $|Q_M^k|$, so $\gamma(S^k) < \gamma(Q_M^k)$. In either case, $\gamma(S^k) < \gamma(Q_M^k)$, which violates the assumption that the set S^k is the CDS with a maximum γ. So Q_M^k must be the induced graph by running ECDS.

Theorem 10: ECDS problem is NP-hard.

Proof: We will prove that the maximum clique problem is Turing reducible to ECDS by constructing a system in which MC can be solved in polynomial time if we can solve ECDS in polynomial time.

Consider the well-known maximum clique problem. Given a graph $G = (V, E)$, find a clique with the maximum number of vertices, where a clique is a sub-graph with all its vertices pairwise adjacent. Now we will construct the system to reduce MC to ECDS.

Given a graph $G = (V, E)$, a complete graph based on G can be defined as $G^+ = (V, E^+ = E \cup E_a)$, where E_a is the set of edges that need to be added to make the original graph G complete. As every vertex in G^+ is pairwise adjacent to others, any clique selected from the constructed complete graph G^+ will be a CDS. This will affect the result of the execution of ECDS in the following steps.

Now we want to find the maximum clique using the following algorithm. The algorithm first finds the maximum clique containing a specific vertex while avoiding using the edges that are not from the original graph G. Once the set of maximum cliques that contains every vertex is constructed, the maximum clique of the original graph can be figured out, e.g., the one with the maximal number of vertices in the set of maximum cliques. The algorithm works as follows.

1) Set $S = V, G = (V, E), Q = \Phi$;
2) Construct G^+;
3) Pick a $v_i \in S$ and let the dominating set $D = \{v_i\}$ and $S = S - \{v_i\}$;
4) Run ECDS on G^+, and construct the dominating set D. Assume $c_i = 1$. For $\forall k \neq i$, set $c_k = \infty$ if $\exists v_k, v_j \in D, (v_k, v_j) \in E_a$, i.e., we want to avoid selecting a node that does not have the edge connection to the constructed D in the original graph G to form D; otherwise let $c_k = \epsilon \rightsquigarrow 1$;
5) Let the graph Q_i be the graph induced by D after the execution of step 4, let $Q = Q \cup \{Q_i\}$;
6) Go to step 3 until $S = \Phi$.

According to Lemma 4, the resulting graph Q_i by running ECDS over the constructed graph G^+, which is a part of the original graph when the cost is set to avoid selecting nodes that do not have an edge connection in the original graph G to the constructed dominating set D, is the maximum clique of G that contains the vertex v_i. As Q is a clique set, with each element being a maximum clique containing a specific vertex of the original graph G, the element in Q that has the largest number of vertices is the maximum clique of G. As all steps in the above algorithm can be executed in polynomial time based on a solution for ECDS, the NP-hard Maximum Clique problem is polynomial reducible to ECDS, which proves that ECDS is NP-hard.

VI. CENTRALIZED ALGORITHM

To obtain an approximate solution for the ECDS problem and construct a reliable and cost effective backbone network, we first consider a centralized reverse greedy (CRG) algorithm as a possible solution to find a CDS of the network graph that has heuristically large connectivity efficiency (CE) γ.

Algorithm 1 CRG

```
BN ← V
for do
    if ¬∃ removable v that γ(BN−v) > γ(BN) then
        return BN
    else
        find removable v to max γ(BN−v)
        BN ← BN −v
    end if
end for
```

In Algorithm 1, CRG forms the backbone network by removing unnecessary nodes from the candidate backbone set, and in each round a node whose removal leads to the maximum increase of CE is removed. The node removing process is repeated until no removal of node could lead to the increase of CE.

Although CRG always removes the node that could lead to the maximum increase of CE in each round, as other greedy algorithms, it may not lead to a globally optimal performance. CRG tends to terminate early at a local optimal point. We further develop a randomized centralized reverse greedy algorithm (RCRG) based on the rules in generic probabilistic meta-algorithm [40]–[42]. The performance shown in Fig. 2 demonstrates the effectiveness of using RCRG. The throughput of RCRG doubles or triples that of CRG at the highest node density and moving speed studied.

In RCRG algorithm shown in Algorithm 2, we introduce a pseudo connectivity efficiency $\zeta(D) = \frac{a^{\beta}(D)}{c(D)}$ to enhance the performance of $\gamma(BN)$ globally. The parameter β is used to control the tradeoff between algebraic connectivity and cost. Generally, we set $\beta \geq 1$ to provide a higher weight to algebraic connectivity. The selection of β also depends on the value ranges of cost $c(D)$ and algebraic connectivity $\alpha(D)$.

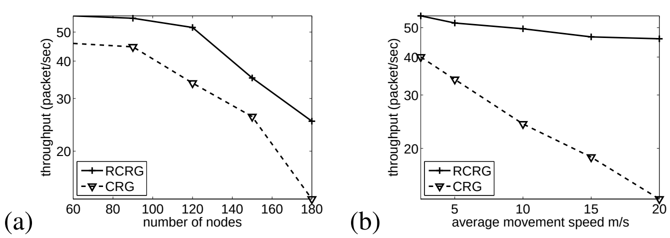

Fig. 2. Simulation comparison between CRG and RCRG: (a) network throughput versus node density; (b) network throughput versus node movement speed.

The algorithm first looks for a candidate set, where a candidate node is the one whose removal from or addition to the current backbone set does not change the connected and dominating property of the backbone set. Let D_i be a graph generated by removing/adding a candidate node i along with the edges incident to it from/into the current graph D. To help better select the backbone nodes, we introduce a facilitating function, $\theta_i = e^{1-\frac{\zeta(D)}{\zeta(D_i)}}$.

The index m indicates the test round and the algorithm begins when the backbone network consists of all the network nodes. In a round, instead of directly removing or adding a node i whose removal or addition leads to the maximum increase in $\zeta(D)$, the backbone node set change has a probability $P(i,T) = max\{min\{(\theta_i^{\frac{1}{T}} - \frac{1}{2}), 1\}, 0\}$ of being made, with $T = \frac{1}{\sqrt{m}}$. This probability is designed to increase with the facilitating function θ_i and decrease with the time function T. The node i whose removal or addition leading to a larger pseudo connectivity efficiency $\zeta(D_i)$ would result in a larger θ_i, and therefore has a higher probability of being removed or added. For more stable performance, we constrain that only one backbone node set change can be made in one round. The reason of removing or adding a 'worse' node i (not leading to the maximum $\zeta(D_i)$) with a probability is to allow the system to move to a new state to prevent the method from being stuck in a local optimum. Based on our definition of $P(i,T)$, the probability of removing or adding in a less optimal node in a round reduces and tends to reach zero when the round index m becomes large. Therefore, RCRG algorithm will converge and become a greedy algorithm (CRG) after a sufficient number of rounds.

Algorithm 2 RCRG

```
D ← V, D' ← V
for m ← 1:mmax do
    if ¬∃i as a candidate then
        return D'
    else
        pick i from candidates
        calculate ζ(D_i)
        if ζ(D_i) > ζ(D') then
            D' ← D_i
        end if
        calculate P(i,T)
        if P(i,T) > uniform() then
            D ← D_i
        end if
    end if
end for
return D'
```

VII. DISTRIBUTED ALGORITHM

With complete network information, a centralized algorithm could provide a better performance. However, a distributed algorithm would be more efficient when the network size is big or the network is more dynamic. In this work, we introduce a distributed algorithm for ECDS problem by leveraging our RCRG algorithm in a distributed environment to form a more reliable and cost effective backbone network. The algorithm can be decomposed into two steps.

Step I. Find Dominating Set.

Our algorithm constructs a dominating set through the finding of maximal independent set (MIS) which selects nodes based on the cost factor, as shown in Algorithm 3. WHITE nodes are the ones that do not belong to any set. In lines 2 to 5, a node with the lowest cost among WHITE neighbors selects itself as a Dominator and announces its status to its one-hop neighbors. In case that more than one neighboring WHITE nodes have the same cost, the one with the highest ID will be selected as the Dominator. In lines 6 to 12, a node receiving the dominator announcement becomes the Dominatee and broadcasts the Dominatee status to its one-hop neighbors, which update the list of WHITE neighbors. A random delay is introduced before each node sends a message to reduce collisions.

Step II. Find Relays.

Algorithm 3 MIS

$V \leftarrow$ WHITE
if $c(u)$ is min in WHITE neighbors or multiple WHITE nodes have the same cost $c(u)$ but u has the largest ID **then**
    u sends MsgDominator up to 1-hop
    u.status $\leftarrow$ Dominator
end if
if v receives MsgDominator **then**
    v.status $\leftarrow$ Dominatee
    v sends MsgDominatee up to 1-hop
end if
if w receives MsgDominatee from v **then**
    w.neighbor(v).status $\leftarrow$ not WHITE
end if

In order to form a CDS of the graph, we need to find some relay nodes to connect the independent set obtained from the first step. Based on [10], if the original graph is connected, a graph $VirtG$ that connects all pairs of elements of a dominating set is a connected graph if there is a path of at most 3 hops in the original graph. Therefore, we connect each pair in the independent set that is within 3-hop distance to form the backbone network by using RCRG algorithm. The CDS formulation procedures are described in Algorithm 4.

For convenience, denote the maximal independent set found in Step I as D.

In lines 1 to 3, each node v in D runs RCRG over the nodes in its 2-hop neighborhood and selects some of the nodes as backbone nodes (BNs), while the remaining nodes are backbone capable nodes (BCNs). A node v announces the results through an RLA message up to two hops, and a node u receiving the message changes its status according to the assignment. If a node u receives conflicting assignments (Bn or BCN) from multiple dominator nodes, it will set its status to BN and announce its status up to two hops. In lines 12 to 16, a node w in $\neg D$ first checks if all the Dominators within its two-hop distance have completed the RCRG calculations. If this process is completed, w checks if two Dominators within 3-hop distance are not connected by backbone nodes, and will change its status to backbone node if there are no-connected Dominators. In lines 17 to 21, unnecessary nodes are removed from the backbone to improve the connectivity efficiency. A higher algebraic connectivity generally helps to improve network stability upon dynamics. In addition, in lines 22 to 26, if a BCN node x finds it loses the connection with all backbone nodes but there is a backbone node two-hops away, it will run RCRG and send other nodes the results through an RLA message. The steps in lines 12 to 16 will also be run by a BCN node to maintain the backbone network connectivity. If there is a significant topology change in a neighborhood, the algorithm may be re-run by resetting all the relevant nodes to white.

Algorithm 4 RELAY

1: **if** $v \in D$ **then**
2:     v runs RCRG over two-hop nodes and sends RLA (BNs, BCNs) up to 2 hops after a random delay
3: **end if**
4: **if** u receives a RLA **then**
5:     u.status $\leftarrow$ BN/BCN based on the assignment in RLA
6:     u.neighbor(v).status $\leftarrow$ assigned
7: **end if**
8: **if** u receives more than one RLA with conflicting status assignment **then**
9:     u.status $\leftarrow$ BN and u sends RLA with its status up to 2 hops after a random delay
10:     u.neighbor(v).status $\leftarrow$ assigned
11: **end if**
12: **if** $w \notin D$ and $\forall$ Dominator in 2 hop assigned **then**
13:     **if** there are non-connected Dominators **then**
14:         w.status $\leftarrow$ BN and w sends RLA up to 2 hops after a random delay
15:     **end if**
16: **end if**
17: **if** x is BN and y is in x's 2 hop **then**
18:     **if** neighbor$(x) \subset$ neighbor(y) **then**
19:         x.status $\leftarrow$ BCN and x sends RLA with its status up to 2 hops after a random delay
20:     **end if**
21: **end if**
22: **if** x is BCN and $\neg\exists x$'s 1 hop BN neighbor **then**
23:     **if** $\exists x$'s 2 hop BN neighbor **then**
24:         x runs RCRG and sends RLA up to 2 hops after a random delay
25:     **end if**
26: **end if**

VIII. Performance Evaluation

In this section, we study the backbone performance by comparing our centralized backbone construction algorithm RCRG and distributed backbone construction algorithm DCRG with two other backbone construction algorithms, (MR-)TSA [7], [23] and k-Coverage [24]. (MR-)TSA is a backbone topology synthesis algorithm based on an abstract weight to construct and maintain a wireless backbone while k-Coverage is an algorithm to construct a wireless backbone which is k-connected and k-dominating. The algorithms are implemented using the network simulator NS2 [16], and the node movement follows the improved random way point model [43]. IEEE 802.11 MAC layer and physical layer models are used, and the transmission range is set at 250 meter. AODV [6] is used as the routing protocol, with the path searching messages RREQ forwarded only by backbone nodes. Each simulation lasts for 180 seconds, and the results are obtained by averaging over five runs. Unless when studying the impact of different parameters, 120 nodes are used in a 1500m x 1500m network area, with the average node moving speed set at

5 m/s. Sixty CBR flows are generated between random sources and destinations, each transmitting at 200 bps. Four main performance metrics, namely throughput, delivery ratio, average end-to-end delay and routing distance, are examined in this study. Throughput is obtained by dividing the total number of packets received at end users by the simulation time, and delivery ratio is calculated by dividing the number of packets received at end users by the total number of packets sent out. Average end-to-end delay is the average duration between the time a packet is sent out and the time the packet is received at the destination, while routing distance is the average number of hops that a packet traverses before it reaches its destination. In implementing (MR-)TSA, BN_Neighbor_Limit is set to 12, h is set to 1. Short_Timer and Long_Timer are 1 and 3 seconds respectively. Generally, the number of backbone nodes will increase significantly when the enforced degree of each node, k, increases. This would lead to higher number of transmission collisions and packet losses, thus reduced network capacity and routing delay. Therefore, in k-Coverage algorithm, k is set to 2 to ensure robustness without including an excessive number of backbone nodes. We study the impact on performance due to network size, node density, network load, and node moving speed. Specific parameter setting will be described in each simulation. In these simulations, as the reference algorithms do not have clear cost models, the cost of each node is randomly generated for both our algorithms and the reference algorithms. We have performed additional simulations to show the benefit of including the cost into backbone control metric using the cost model described in Section IV.

A. Impact of backbone network

We vary node density and movement speed to compare the throughput between a network with and without backbone infrastructure. In Fig. 3(a), the AODV is run over the original network, the throughput is higher when the network is sparse and has a low control overhead, but the throughput reduces dramatically when the node density increases. The use of backbone infrastructure greatly reduces the number of path discovery messages and collisions, and DCRG gains more than double the throughput of AODV at the maximum node density tested. From Fig. 3(b), we can see that a network with backbone has much more stable throughput when node mobility increases, and the throughput of DCRG is about three times that of AODV at node speed 20m/s. This is because the backbone network proposed is constructed to improve algebraic connectivity, and a higher algebraic connectivity would ensure more reliable transmissions in a dynamic environment [20]. Additionally, the higher probability of link failure due to mobility would lead to an increase of route recovery messages and thus a larger number of collisions and higher throughput reduction.

B. Impact of Metric

A good metric is important for backbone construction and quality. The objective of our backbone algorithm is to optimize the connectivity efficiency, which is a function of algebraic connectivity and cost. We introduce a cost model in Section IV, to help improve network performance by selecting backbone nodes based on node distribution, traffic load and hence errors, and node capacity. Due to the page limit, we only show the impacts due to node distribution and load, with the unbalance level of each controlled through a standard deviation from 0 to 4. The results in Fig. 4 (a) and (b) show the performance of using the metric with algebraic connectivity and a random cost (RCRG, DCRG), and the metric with algebraic connectivity and the cost model introduced (RCRG-C, DCRG-C). Our results show that using an effective cost model could lead to an increased throughput, about 20% in this simulation. The performance improvement is higher when the network is moderately unbalanced, while the improvement reduces if the unbalanced level is too big, as the later dominates the network performance. Improvements are also observed when varying the node density and speed in Fig. 4 (c) and (d). In the next several sections, we are going to show the performance using a relatively balanced topology and random cost, to mainly evaluate the performance impact due to algebraic connectivity.

C. Impact of network size

We vary the network size from 1000m x 1000m to 2000m x 2000m, while fixing the network density at 53 nodes / km^2. In Fig.5 (a) and (b), both network throughput and delivery ratio decrease with network size for all the algorithms, as the increase of average path length (Fig.5 (d)) results in a higher probability of packet collision and therefore loss. Both RCRG and DCRG are seen to perform much better than TSA and k-Coverage at all network sizes. Compared to k-Coverage, RCRG has up to 100% higher throughput and delivery ratio, while DCRG has up to 60% performance improvement. TSA has the lowest throughput and delivery ratio as a result of backbone bottlenecks. In Fig.5 (c) and (d), both average end-to-end delay and average routing distance are observed to increase with network size. RCRG and DCRG have lower end-to-end delay with the use of more efficient routing paths. DCRA has up to 60% lower delay as compared to k-Coverage, and up to 70% lower delay as compared to TSA.

TSA intends to have a backbone network with a smaller number of nodes and lower cost to reduce transmission collisions and increase network throughput, however, this can create bottlenecks in the backbone network. As the network size increases, this probability also increases, and the performance is greatly impacted by these bottlenecks. On the other hand, targeted for a higher reliability, k-Coverage is too conservative by ensuring 2-connectivity for each backbone node, which leads to a larger number of backbone nodes and hence more collisions in transmissions. Both RCRG and DCRG use algebraic connectivity as part of the design metric to ensure the backbone network to be more robust and to increase bottleneck capacity, and the routing path to be more efficient. As a result,

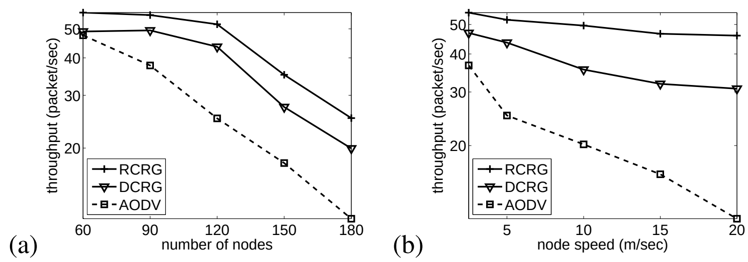

Fig. 3. Impact of Backbone Network: (a) network throughput versus node density; (b) network throughput versus node movement speed.

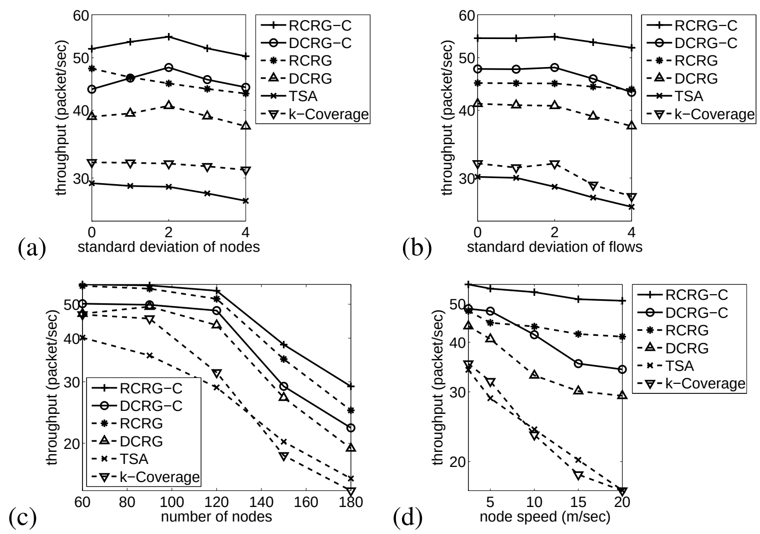

Fig. 4. Impact of Metric: (a) network throughput versus standard deviation of node distribution; (b) network throughput versus standard deviation of load distribution; (c) network throughput versus node density; (d) network throughput versus node movement speed.

these two algorithms have shorter transmission distance, lower transmission delay, higher delivery ratio and higher throughput.

D. Impact of Network Load

We evaluate the impact of network load on the performance by varying the number of flows from 30 to 90 with each flow transmitted at 200 bps. In Fig.5 (a), the throughput of all the algorithms is observed to increase with the network load initially and then goes down after the network is saturated by the load. RCRG and DCRG can support about 30% higher network load compared to TSA and k-Coverage, with a higher backbone capacity and more balanced transmissions. In Fig.6 (b), the delivery ratio drops quickly with the increase of load as a result of collision and congestion. The throughout improvement of RCRG and DCRG compared to TSA and k-Coverage increases as the load increases, and DCRG has about 50% higher throughput at the highest load tested. TSA has a relatively lower throughput compared to k-Coverage when the load is low due to the bottleneck impact of the backbone network, but outperforms k-Coverage when the collision begins to dominate at a high network load. For all the algorithms, the average end-to-end delays are seen to increase as the load becomes heavier in Fig.6 (c), while the average routing distances remain stable in Fig.6 (d). The end-to-end delays increase because of the increase in the number of retransmissions and queuing delay. As the network load does not have a significant impact on the topology, the average routing distances do not have big changes. The slight reduction of routing distance at high load is due to the higher probability of dropping the packets that have a longer transmission path. With the consideration of algebraic connectivity, RCRG and DCRG have shorter routing paths and hence lower delay than TSA and k-Coverage, which do not consider the bounding of the routing path. The lower delays of RCRG and DCRG are also due to their higher backbone capacity and their considerations of load balancing.

E. Impact of Node Density

We keep the network size at 1500 m x 1500 m, and vary the number of nodes from 60 to 180. In Fig.7 (a) and (b), the throughput and delivery ratio reduce as the node density increases, due to a larger probability of transmission collisions. Although there is only a small increase in the number of backbone nodes, the transmissions from regular nodes also create collisions. We observe that RCRG and DCRG outperform k-Coverage by 40% and 30% respectively on average, and outperforms TSA by 45% and 35% respectively on average. In a relatively low density network, k-Coverage performs well due to the 2-connectivity

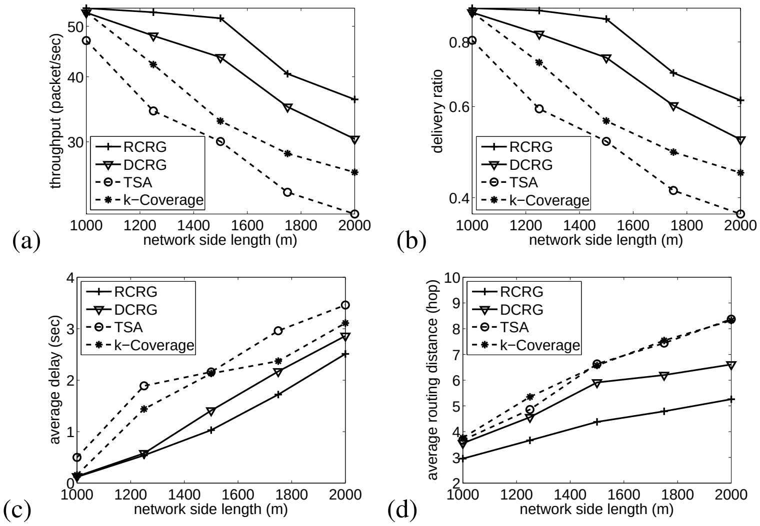

Fig. 5. Impact of Network Size: (a) network throughput; (b) delivery ratio; (c) average end-to-end delay; (d) average routing distance.

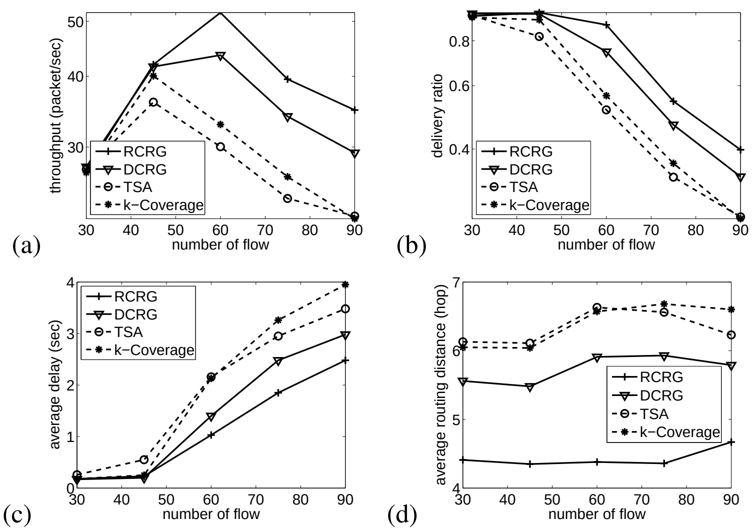

Fig. 6. Impact of Network Load: (a) network throughput; (b) delivery ratio; (c) average end-to-end delay; (d) average routing distance.

2-dominating robustness. However, when the density of the network increases, the transmission collisions of k-Coverage become serious due to its larger number of backbone nodes, which makes its throughout and delivery ratio drop quickly and even below TSA at high node density. In Fig.7 (c) and (d), we can see that the delays of RCRG and DCRG are about 40-50% lower than that of TSA and k-Coverage. At very low density, k-Coverage has a short delay, but the delay quickly increases as the network density increases. With the increase of network density, the routing distances of RCRG and DCRG remain stable, while the average path lengths of TSA and k-Coverage increase quickly. This again shows the effectiveness of exploiting the algebraic connectivity to control routing distance.

F. Impact of Node Mobility

One of the major goals of our algorithms is to improve network reliability. In this simulation, we study the impact on performance due to mobility and the resulting topology change. We vary the average node moving speed from 2.5 m/s to 20 m/s. Fig.8 shows that TSA and k-Coverage have similar throughput and delivery ratio, which reduce quickly as the nodes move faster. RCRG and DCRG have much more stable performance. The difference between the throughput and delivery ratio of RCRG/DCRG and TSA/k-Coverage increases as the node mobility increases. At the maximum speed tested, DCRG has about 60% higher throughput and delivery ratio than that of TSA and k-Coverage. TSA attempts to maintain the backbone network when the network topology changes, and k-Coverage is designed to support higher backbone reliability. The significant performance improvements of RCRG and DCRG demonstrate the effectiveness of using algebraic connectivity to support more robust network design. With the increase of mobility, the end-to-end delay of TSA and k-Coverage increase much faster than that of RCRG ad

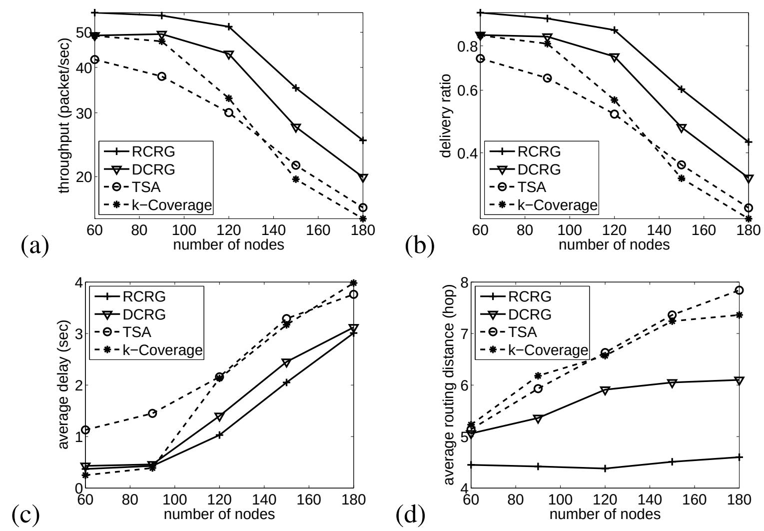

Fig. 7. Impact of Node Density: (a) network throughput; (b) delivery ratio; (c) average end-to-end delay; (d) average routing distance.

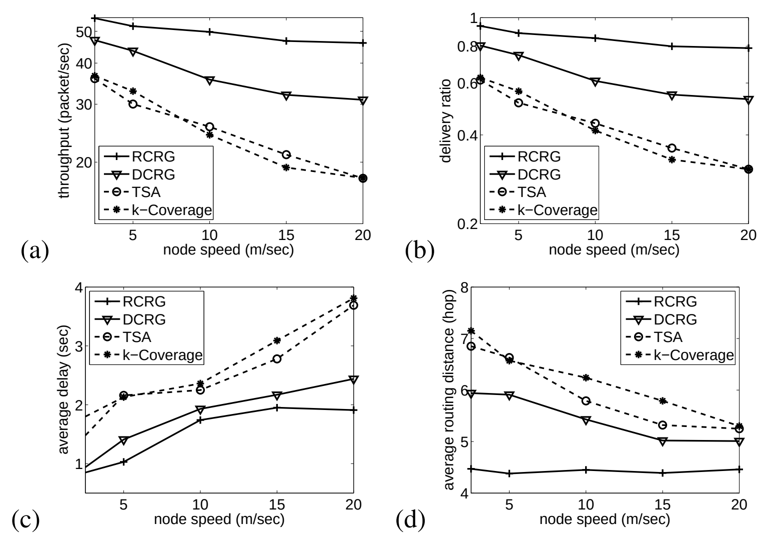

Fig. 8. Impact of Node Mobility: (a) network throughput; (b) delivery ratio; (c) average end-to-end delay; (d) average routing distance.

DCRG due to the increase of link breakages, retransmissions, and routing path re-establishments. The routing distances of TSA, k-Coverage and DCRG all reduce, as a long path transmission has a much higher probability of failure than a short path transmission. RCRG and DCRG both have relatively lower delay and shorter routing distance.

IX. Conclusions

With the increasing demand of wireless network applications, it is critical to develop more effective communications paradigm to enable new and powerful pervasive applications. To cope with the increase in the number of communication devices, many efforts have been made in recent years to improve network throughput by constructing a minimum-size backbone network to reduce total network transmissions and hence collisions. However, wireless network throughput is also impacted by bottleneck network flow rate, and transmission distance. It is also important to consider backbone reliability, stability, and load balancing.

In this work, we exploit the use of algebraic connectivity to capture the spectral characteristics of the network graph in our backbone design to simultaneously improve backbone network robustness, capacity, stability, and routing efficiency. In order to meet different application needs, we introduce a general cost function to incorporate other desired network features. We define a new metric, connectivity efficiency, to tradeoff algebraic connectivity and cost during backbone formulation. As a design example, we provide a cost function to capture the impact of node bandwidth and transmission errors, and to balance the network load based on node distributions. This is the first work that comprehensively considers all the desired network features in constructing the backbone.

We formally formulate our backbone construction problem as the connected dominating set (CDS) problem by selecting a subset of nodes from backbone capable nodes to form a connected dominating set, with the objective of maximizing network connectivity efficiency. We prove that the connectivity efficiency maximization problem is NP-hard, and propose a centralized and a distributed approximation algorithms to solve the problem. Finally, we perform simulations to compare the performance of our algorithms and algorithms proposed in the literature. Our performance studies demonstrate the effectiveness of using algebraic connectivity as the performance metric in constructing the backbone network. Compared to peer algorithms, our algorithms have much higher throughput and delivery ratio, and much lower end-to-end delay and routing distances under all test scenarios, including the network size, node density, network load, and node mobility.

REFERENCES

[1] Jie Wu, Fei Dai, and Shuhui Yang. Iterative Local Solutions for Connected Dominating Set in Ad Hoc Wireless Networks. *IEEE Trans. on Computers*, 57(5), Page(s):702 - 715, May 2008.

[2] Y. C. Tseng, S. Y. Ni and E. Y. Shih. *"Adaptive Approaches to Relieving Broadcast Storms in a Wireless Multihop Mobile Ad Hoc Network,"* IEEE Transactions on Computers, Vol. 52, No. 5, pp. 545-557, May 2003.

[3] Y. Cai, W. Lou, M. Li and X.-Y Li. *"Energy Efficient Target-Oriented Scheduling in Directional Sensor Networks,"* IEEE Transactions on Computers, Vol. 58, No. 9, pp. 1259 -1274, September, 2009,

[4] X. Xiang, X. Wang and Y. Yang. *"Stateless Multicasting in Mobile Ad Hoc Networks,"* IEEE Transactions on Computers, to appear.

[5] W. Wu, J. Cao, J. Yang and M. Raynal. *"Design and Performance Evaluation of Efficient Consensus Protocols for Mobile Ad Hoc Networks,"* IEEE Transactions on Computers, Vol. 56, No. 8, pp. 1055 -1070, Aug, 2007.

[6] C. E. Perkins, E. M. Royer. *"Ad hoc On-Demand Distance Vector Routing,"* Proc. 2nd IEEE Workshop on Mobile Computing Systems and Appl., pp. 90-100, February 1999,

[7] H. Ju, I. Rubin, K. Ni, C. Wu, *"A Distributed Mobile Backbone Formation Algorithm for Wireless Ad Hoc Networks,"* BROADNETS 04, pp. 661-670, Apr. 2004.

[8] H.F. Weinberger, *"Variational Methods for Eigenvalue Approximation,"* SIAM, 1974.

[9] G. Kirchhoff, *"uber die Auflosung der Gleichungen, auf welche man bei der untersuchung der linearen verteilung galvanischer Strome gefuhrt wird,"* Ann. Phys. Chem. 72, 497-508, 1847.

[10] K. M. Alzoubi, P. -J. Wan, O. Frieder, *"New Distributed Algorithm for Connected Dominating Set in Wireless Ad Hoc Networks,"* Proc. IEEE Hawaii Int'l Conf. System Sciences, 2002.

[11] J. Wu and H. Li, *"On Calculating Connected Dominating Set for Efficient Routing in Ad Hoc Wireless Networks,"* Proc. Third Int'l Workshop Discrete Algo. and Methods for Mobile Comp. and Comm., 1999.

[12] F. Dai, J. Wu, *"An Extended Localized Algorithm for Connected Donminating Set Formation in Ad Hoc Wireless Networks,"* IEEE Trans. Paral. and Distributed Systems, vol 15, no. 10, pp. 908-920, Oct. 2004.

[13] Y. Wang, W. Wang, X. Li, *"Efficient Distributed Low-Cost Backbone Formation for Wireless Networks,"* IEEE Trans. Parallel and Distributed Systems, vol 17, no. 7, pp. 681-693, Jul. 2006.

[14] B. Mohar, *"The Laplacian Spectrum of Graghs,"* Graph Theory, Combinatorics, and Applications, Vol. 2, 871-898, 1991.

[15] M. Fiedler, *"Algebraic Connectivity of Graphs,"* Czechoslovak Mathematical Journal, 23:298-305, 1973.

[16] *"The Network Simulator - ns-2,"* http://www.isi.edu/nsnam/ns/.

[17] Y. Wu and Y. Li, *"Construction Algorithms for k-Connected m-Dominating Sets in Wireless Sensor Networks,"* Proc. ACM Mobihoc 2008, May 2008

[18] D. M. Blough, M. Leoncini, G. Resta, P. Santi, *"The k-Neighbors Approach to Interference Bounded and Symmetric Topology Control in Ad Hoc Networks,"* IEEE Trans. on Mobile Computing, Vol. 5, n. 9, pp. 1267-1282, Sept. 2006

[19] C. Scheideler, A. Richa, P. Santi, *"An O(log n) Dominating Set Protocol for Wireless Ad Hoc Networks under the Physical Interference Model,"* Proc. ACM MobiHoc, pp. 91-100, 2008

[20] J. Alexander Fax and Richard M. Murray, *"Information Flow and Cooperative Control of Vehicle Formations,"* IEEE Transaction on Automatic Control, Vol. 49, No. 9, September 2004.

[21] N. Alon, V. D. Milman, *"λ_1, isoperimetric inequalities for graphs and superconecntrators,"* J. Combinatorics Theory, Ser. B38, 73-88, 1985.

[22] Eugene Lawler, Combinatorial Optimization: Networks and Matroids. Dover Publications. pp. 117-120, 2001.

[23] H. Ju, I. Rubin *"Backbone Topology Synthesis for Multi-Radio Meshed Wireless LANs,"* INFOCOM 2006., pp. 1-12, April 2006.

[24] F. Dai, J. Wu, *"On Constructing k-Connected k-Dominating Set in Wireless Ad Hoc and Sensor Networks."* J. Paral. and Distributed Comp., vol.66, no.7, pp. 947-958, July 2006.

[25] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris, *"Span: An Energy-Efficient Coordination Algorithm for Topology Maintenance in Ad Hoc Wireless Networks,"* Wireless Network, vol. 8, no. 5, pp. 481-494, 2002.

[26] S. Basagni, *"Finding a Maximal Weighted Independent Set in Wireless Networks,"* Telecomm. Systems, vol. 18, nos. 1-3, pp. 155-168, Sept. 2001.

[27] C. Bettstetter and R. Krausser, *"Scenario-Based Stability Analysis of the Distributed Mobility-Adaptive Clustering (DMAC) Algorithm,"* Proc. Second ACM Int'l Symp. Mobile Ad Hoc Networking and Computing, 2001.

[28] U.C. Kozat, G. Kondylis, B. Ryu, and M. Marina, *"Virtual Dynamic Backbone for Mobile Ad Hoc Networks,"* Proc. IEEE Int'l Conf. Comm., 2001.

[29] M. Gerla and J.T.-C. Tsai, *"Multicluster, Mobile, Multimedia Radio Network,"* Wireless Networks, vol. 1, no. 3, pp. 255-265, 1995.

[30] C.R. Lin and M. Gerla, *"Adaptive Clustering for Mobile Wireless Networks,"* IEEE J. Selected Areas in Comm., vol. 15, no. 7, pp. 1265-1275, Sept. 1997.

[31] S. Basagni, M. Mastrogiovanni, A. Panconesi and C. Petrioli, *"Localized Protocols for Ad Hoc Clustering and Backbone Formation: A Performance Comparison,"* IEEE Trans. on Parallel and Distributed Systems, Vol. 17, No. 4, April 2006.

[32] A. Srinivas, G. Zussman and E. Modiano, *"Mobile Backbone Networks - Construction and Maintenance,"* MobiHoc'06, May 2006.

[33] P. Wei, S. Chu and X. Wang, *"Deployment of a Reinforcement Backbone Network with Constraints of Connection and Resources,"* IEEE International Conference on Distributed Computing Systems, Genoa, Italy, June, 2010.

[34] M.V. Marathe, H. Breu, H.B. Hunt III, S.S. Ravi, and D.J. Rosenkrantz, *"Simple Heuristics for Unit Disk Graphs,"* Networks, vol. 25, pp. 59-68, 1995.

[35] I. Rubin, A. Behzad, R. Zhang, H. Luo, and E. Caballero, *"TBONE: a Mobile-Backbone Protocol for Ad Hoc Wireless Networks,"* Proc. IEEE Aerospace Conf., March 2002.

[36] D. Kim, Y. Wu, Y. Li, F. Zou, and D. Du, *"Constructing Minimum Connected Dominating Sets with Bounded Diameters in Wireless Networks,"* IEEE Transactions on Parallel and Distributed Systems (TPDS), vol. 20, no. 2, pp. 147-157, February, 2009.

[37] D. Kim, Y. Wu, Y. Li, F. Zou, and D. Du, *"Trade-Off Scheme for Fault Tolerant Connected Dominating Sets on Size and Diameter,"* Proceedings of ACM International Workshop on Foudations of Wireless Ad Hoc and Sensor Networking and Computing (FOWANC), 2008.

[38] K. Xu, X. Hong, and M. Gerla, *"Landmark routing in Ad Hoc Networks with Mobile Backbones,"* J. Parallel Distrib. Comput., 63(2):110-122, 2003.

[39] H. Ju and I. Rubin, *"Performance Analysis and Enhancement for Backbone Based Wireless Mobile Ad Hoc Networks,"* Proc. IEEE BROADNETS Conference, Boston, MA, October 2005.

[40] S. Kirkpatrick and C. D. Gelatt and M. P. Vecchi, *"Optimization by Simulated Annealing,"* Science, Vol 220, Number 4598, pages 671-680, 1983.

[41] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, and E. Teller, *"Equations of State Calculations by Fast Computing Machines,"* Journal of Chemical Physics, 21(6):1087-1092, 1953.

[42] V. Cerny, *"A thermodynamical approach to the travelling salesman problem: an efficient simulation algorithm,"* Journal of Optimization Theory and Applications, 45:41-51, 1985.

[43] W. Navidi and T. Camp, *"Stationary Distributions for the RandomWaypoint Mobility Model,"* IEEE Trans. on Mobile Computing, vol. 3, no. 1, pp. 99-108, 2004.

[44] H. Wang and P. Van, *"Algebraic connectivity optimization via link addition,"* International Conference On Bio-Inspired Models Of Network, Information And Computing Systems, 2008.

[45] Damon Mosk-Aoyama, *"Maximum algebraic connectivity augmentation is NP-hard,"* Operations Research Letters, Vol. 36, Issue 6, Nov 2008, pp. 677-679.