Exploiting Use of a New Performance Metric for Construction of Robust and Efficient Wireless Backbone Network

Ziyi Zhang, Qiang Ma, and Xin Wang
Department of Electrical and Computer Engineering
Stony Brook University
Stony Brook, New York, 11794
Email: {zizhang, qiangma, xwang}@ece.sunysb.edu

***Abstract*— In order to improve transmission throughput of a multi-hop wireless network, many efforts have been made in recent years to reduce traffic and hence transmission collisions by constructing backbone networks with 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 also provide a general cost function and introduce a new metric, connectivity efficiency, to tradeoff algebraic connectivity and cost for backbone construction. We have designed both centralized and distributed algorithms to build more robust and efficient backbone infrastructure to better support the application needs. Our performance studies demonstrate that, compared to peer work, our algorithms could achieve much higher throughput and delivery ratio, and much lower end-to-end delay and routing distances under all test scenarios.**

I. INTRODUCTION

It is extremely challenging to support efficient and reliable communications over multi-hop wireless networks, due to node mobility, device unreliability and unstable wireless communications medium. The increase of network nodes as a result of quick growth of wireless devices and communication need introduces additional challenge to wireless network design. Different from wired networks where a larger number of network nodes could potentially lead to the increase of throughput and reduction of network diameter, due to the sharing of transmission medium, the increase of node competition in channel access would lead to more transmission collisions hence higher transmission delay, throughput degradation, and extra energy consumption. Therefore, backbone design has received significant attentions in recent years to improve wireless network performance. Existing work, however, mainly focuses on minimizing the total number of backbone nodes [4]–[6], [16], [18]–[22]. Many other important issues need to be considered in backbone design, such as backbone network reliability, stability, capacity, load balancing, path length, energy level and therefore longevity. The limited work considering backbone transmission reliability normally try to ensure certain degree connectivity around each node, without considering network capacity and routing distance. Maintaining a fixed degree of connectivity for each node tends to be conservative, which unnecessarily incorporates more backbone nodes and may reduce backbone throughput.

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. 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.

Instead of simply covering all RNs [26], we intend to increase backbone robustness in presence of node mobility, device unreliability and channel instability, while considering other desired network features. Specifically, we exploit the use and control of *algebraic connectivity* [9], an important concept introduced in spectral graph theory, in backbone network design to improve backbone robustness, stability, capacity and routing efficiency. To further capture other network features, we consider the use of a cost function. We further introduce a new metric, *connectivity efficiency*, as a function of algebraic connectivity and total network cost. The purpose of our backbone design is to maximize network connectivity efficiency. This metric 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 exploit 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. We expect the new performance metric proposed in this work can be used in other network research to design high performance network architecture.

We prove that the connectivity efficiency maximization problem is NP-hard, and propose both centralized and distributed approximate algorithms to solve the problem. 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 the node capacity, transmission error, and node distribution. The total network cost is the summation of node cost. Our proposed performance metric and backbone construction algorithms, however, do not constrain the cost function format so that the proposed backbone formulation algorithms can be applied to meeting application needs. Finally, we perform extensive simulations to compare the performance of our algorithms and algorithms proposed in the literature. As this work focuses on backbone construction, we assume all the nodes are backbone capable nodes, i.e., BCNs, and we do not specially identify BCNs in the remaining of the paper.

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

II. RELATED WORK

Cluster organization has been widely studied in literature work and 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 [24], maximum node degree [23], or some generic weight [18]. Ju et al. [15] introduced heuristic approaches to construct the backbone network.

Distributed algorithms to construct connected dominating sets (CDS) in ad hoc networks are studied in [4], [6], [7], [16], [18]. Alzoubi et al. [6] models the transmission range as unit disk, and proposes a localized approximate method to construct minimum CDS within a constant time using a linear number of messages, while the algorithm in [7] reduces the size of the CDS. Marathe et al. [27] also models the network as unit disk graph, and considers methods for constructing maximum independent set, minimum coloring, and minimum dominating set. Chen et al. [18] proposes a localized CDS building algorithm where a node becomes a dominator when two of its neighbors can not reach each other either directly or through one or two dominators. The algorithm in [16] 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. [4] further improve the algorithms proposed in [16]. A survey and simulation-based performance studies were carried in [25] to compare various backbone construction schemes proposed in literature. Scheideler et al. [12] further explored interference model in dominating set problem. Dai et al. and Wu et al. [11], [17] and Zhang et al. [29] considered robust wireless backbone which was k-connected, k-dominating and k-connected, m-dominating respectively. Instead of enforcing conservative vertex degree constraint, the use of algebraic connectivity can better describe the reliability of networks at a large range and the importance of increasing bottleneck capacity.

Algorithms in [5], [18]–[22] also considered using different weights as priority criteria to select clusterheads, while the goal of the majority of the schemes is still 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 [20], and to nodes relatively stable and with high degree in [21]. Basagni gives an algorithm to solve the maximal weighted independent set problem in [19]. Wang et al [5] develops a distributed heuristic algorithm for constructing the minimum weighted dominating set and the minimum weighted connected dominating set. However, these algorithms do not consider the overall backbone network reliability.

The idea of constructing hierarchical backbone was recently considered in [26], [28], [30]. Xu et al. [30] simply selects the nodes that first claim the leadership in a neighborhood to be clusterheads, while TBONE proposed in [28] attempts to minimize the number of backbone nodes, giving priority to higher weight nodes. Work in [26] 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.

III. THEORETICAL FOUNDATION

In this section, we analyze the properties of algebraic connectivity which are important for network design. 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 [9], Fiedler coined algebraic connectivity as $a(G) = \lambda_2$ which is a non-negative real number.

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 distance between all pairs of vertices, and $\overline{\rho}$ represents the average distance. To justify that $a(G)$ is a good measure of graph connectivity, Fiedler and Weyl [2], [9] 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. 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 is added. 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 unless it is critical. In [9], Fiedler provided the following theory.

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

This theorem provides bounds and relates $a(G)$ to the conventional connectivity measures $v(G)$ and $e(G)$. For a network to be reliable, it is desirable to have higher edge and/or vertex connectivity in order to handle link or node failure. This is particularly important for mobile wireless networks. The following theorems proposed by Kirchhoff [3], Alon and Milman [14] 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, it serves as a lower bound of $t(G)$. Theorems 5 and 6 indicate that more edges are 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 and may have a higher capacity according to the max flow theorem. Some recent discoveries by Mohar [8] 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 minor influence on stability for a network with large algebraic connectivity [13].

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 continuous value and provides a fine metric to measure network connectivity level. It not only captures network connectivity, but also to some extent, reflects network stability and gives lower bounds on the performance of the network bottleneck. Additionally, algebraic connectivity controls the upper bound of network routing distance. Thus 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. The effectiveness of using algebraic connectivity for improving network reliability and reducing routing distance is demonstrated through our performance studies in Section VII.

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.

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 $C(G) = \sum_{v_i \in V} c_i$. To improve the robustness, capacity, and efficiency of a graph G while reducing its total cost, we define a new metric called *connectivity efficiency* (CE) as

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

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

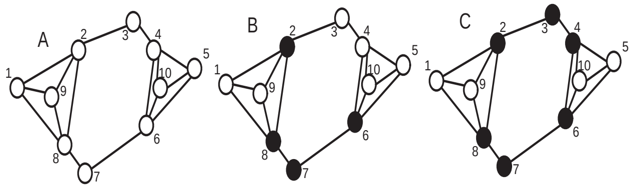


Fig. 1. Example Backbone Construction

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 shown 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. [5]) will, for example, select $\{2, 6, 7, 8\}$ as backbone nodes (marked dark in figure) with 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. With the probability of 0.27 and 0.4, simultaneous failures from two nodes will not impact the function and connection of the backbone.

From Fig. 1B, we also observe that the worst routing path between two nodes has 5 hops while the shortest path between these two nodes has only 3 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. 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.

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 a vertex (i.e., a node), not any edge. ECDS problem is NP-hard, as the NP-hard maximum clique (MC) problem can be polynomially reduced to ECDS.

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

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

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 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 and to achieve a higher network throughput.

V. 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

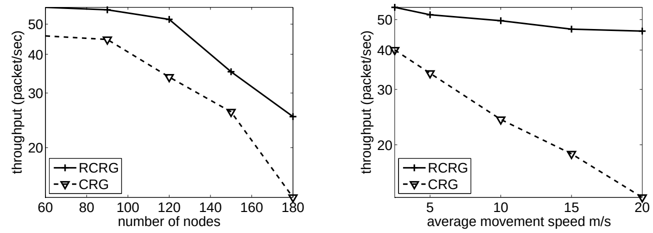
```
BN ← V
for do
    if ¬∃ removable v that γ(BN−v) > γ(BN) then
        return BN
    else
        find removable v to max γ(BN−v)
        BN ← BN −v
    end if
end for
```

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 [31]–[33]. 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 $\alpha(D)$.

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, $\theta_i = e^{1-\frac{\zeta(D)}{\zeta(D_i)}}$.



(a) (b)

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

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.

VI. 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 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) with use of cost factor to select the nodes, 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 lines 6 to 12, a node receiving the dominator announcement becomes the Dominatee and broadcasts the

Algorithm 2 RCRG

1: $D \leftarrow V$, $D' \leftarrow V$
2: **for** m $\leftarrow$ 1:mmax **do**
3:     **if** $\neg\exists i$ as a candidate **then**
4:         **return** D'
5:     **else**
6:         pick i from candidates
7:         calculate $\zeta(D_i)$
8:         **if** $\zeta(D_i) > \zeta(D')$ **then**
9:             $D' \leftarrow D_i$
10:         **end if**
11:         calculate $P(i,T)$
12:         **if** $P(i,T) >$ uniform() **then**
13:             $D \leftarrow D_i$
14:         **end if**
15:     **end if**
16: **end for**
17: **return** D'

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.

Algorithm 3 MIS

1: $V \leftarrow$ WHITE
2: **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**
3:     u send MsgDominator up to 1-hop
4:     u.status $\leftarrow$ Dominator
5: **end if**
6: **if** v receives MsgDominator **then**
7:     v.status $\leftarrow$ Dominatee
8:     v send MsgDominatee up to 1-hop
9: **end if**
10: **if** w receives MsgDominatee from v **then**
11:     w.neighbor(v).status $\leftarrow$ not WHITE
12: **end if**

Step II. Find Relays.

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 [6], 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. 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 receiving the message changes its status according to the assignment. In lines 8 to 10, a node w in $\neg D$ first checks if all the Dominators within 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 13 to 17, unnecessary nodes are removed from the backbone to improve the connectivity efficiency. A higher algebraic connectivity generally helps improve network stability upon dynamics. In addition, in lines 18 to 22, if a BCN node x finds it loses 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 8 to 12 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
3: **end if**
4: **if** u receives 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** $w \notin D$ and $\forall$ Dominator in 2 hop assigned **then**
9:     **if** there are non-connected Dominators **then**
10:         w.status $\leftarrow$ BN
11:     **end if**
12: **end if**
13: **if** x is BN and y is in x's 2 hop **then**
14:     **if** neighbor(x) $\subset$ neighbor(y) **then**
15:         x.status $\leftarrow$ BCN
16:     **end if**
17: **end if**
18: **if** x is BCN and $\neg\exists x$'s 1 hop BN neighbor **then**
19:     **if** $\exists x$'s 2 hop BN neighbor **then**
20:         x runs RCRG and sends RLA
21:     **end if**
22: **end if**

VII. 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 [15] and k-Coverage [17]. (MR-)TSA is a backbone topology synthesis algorithm based on abstract weight to construct and maintain 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 [10], and the node movement follows the improved random way point model [34]. IEEE 802.11 MAC layer and physical layer models are used, and the transmission range is set at 250 meter. AODV [1] 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. In k-Coverage algorithm, k is set to be 2 to ensure robustness without having 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 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 page limit, we only show the impacts due to node distribution and load, with the unbalance level of each controlled through standard deviation from 0 to 4. The results in Fig. 3 (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

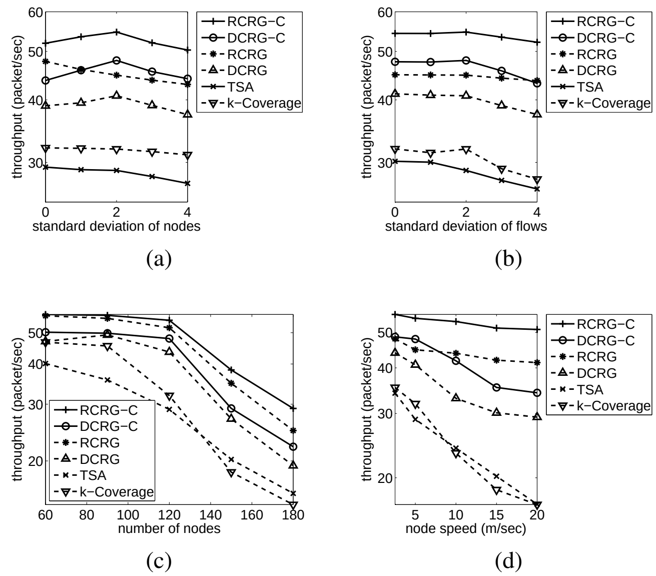


Fig. 3. Impact of Metric: (a) network throughput versus standard deviation of node distribution; (b) network throughput versus standard deviation of load distribution; (c) network throughput versus node density; and, (d) network throughput versus node movement speed.

lead to 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 node density and speed in Fig. 3 (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.

B. 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.4 (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.4 (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.4 (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

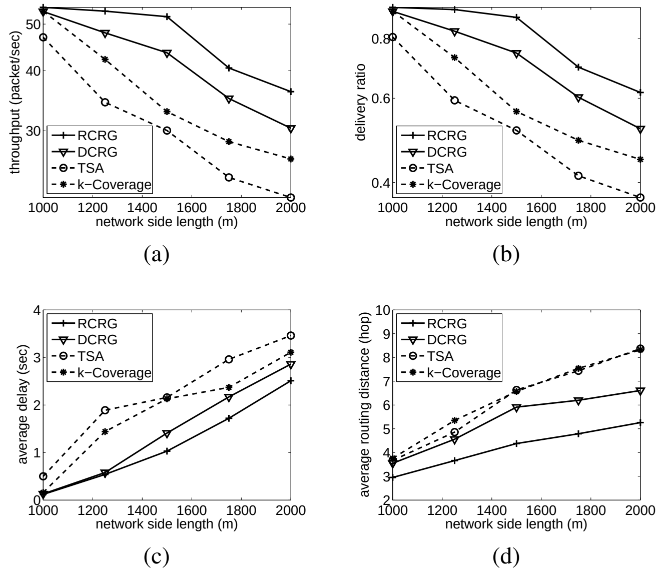


Fig. 4. Impact of Network Size: (a) network throughput; (b) delivery ratio; (c) average end-to-end delay; and, (d) average routing distance.

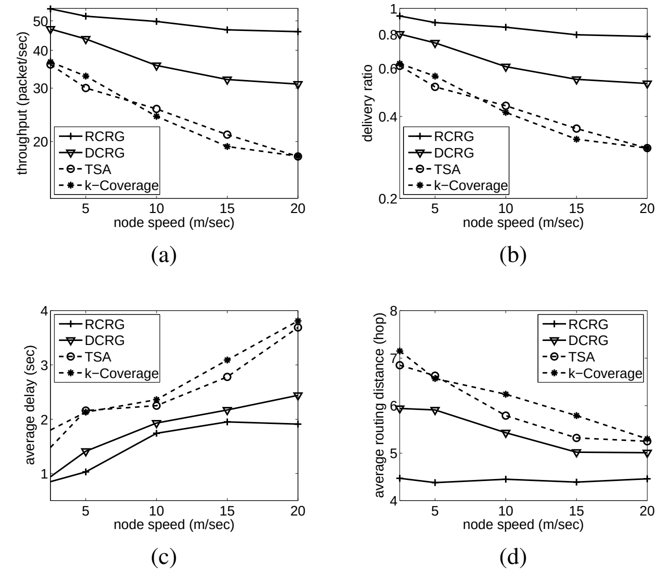


Fig. 5. Impact of Node Mobility: (a) network throughput; (b) delivery ratio; (c) average end-to-end delay; and, (d) average routing distance.

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

C. 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.5 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 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.

VIII. 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 also 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. However, our performance metric and our algorithm design are not constrained by a specific cost function. This is the first work that exploits use of algebraic connectivity and comprehensively considers all the desired network features in constructing the backbone.

The objective of our backbone design is to maximize network connectivity efficiency. We 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 show that the backbone networks constructed using our algorithms are much more robust and efficient than those constructed using literature algorithms. 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. The performance studies demonstrate the effectiveness of using algebraic connectivity in network design.

REFERENCES

[1] 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,

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

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] Xiuzhen Cheng, Min Ding, David H. Du, and X. Jia, *"Virtual Backbone Construction in Multihop Ad Hoc Wireless Networks,"* Wireless Communications and Mobile Computing, Vol. 6, pp. 183-190, 2006.

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

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

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

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

[12] 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

[13] 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.

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

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

[16] 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.

[17] 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.

[18] 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.

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

[20] 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.

[21] 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.

[22] M. Min, F. Wang, D.-Z. Du, and P.M. Pardalos, *"A Reliable Virtual Backbone Scheme in Mobile Ad-Hoc Networks,"* Proc. First IEEE Int'l Conf. Mobile Ad-Hoc and Sensor Systems, 2004.

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

[24] 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.

[25] 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.

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

[27] 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.

[28] 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.

[29] Yiwei Wu and Yingshu Li, Construction Algorithms for k-Connected m-Dominating Sets in Wireless Sensor Networks, Mobihoc 2008, Hong Kong, China, May 26-30, 2008.

[30] 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.

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

[32] 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.

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

[34] 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.