Improving the Topological Resilience of Mobile Ad Hoc Networks

Tae-Hoon Kim, David Tipper, Prashant Krishnamurthy
Graduate Networking and Telecommunications Program
University of Pittsburgh, Pittsburgh PA, USA
Email: tak11@pitt.edu, {dtipper, prashant}@pitt.edu

A. Lee Swindlhurst
Department of Electrical Engineering & Computer Science
University of California, Irvine CA, USA
Email: swindle@uci.edu

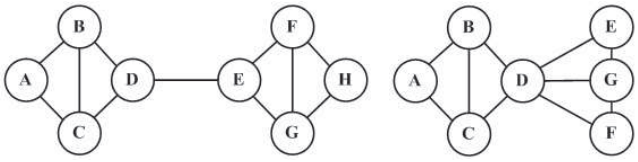
***Abstract*—In order to effectively deploy survivability techniques to improve the resilience of mobile ad hoc networks, one must be able to identify all the weak points of the network topology. The weak or critical points of the topology are those links and nodes whose failure results in partitioning of the network. Here we propose a new algorithm based on results from algebraic graph theory, that can find the critical points in the network for single and multiple failure cases. Utilizing this algorithm we present numerical results that examine how the number of critical points varies with nodal density. Secondly, we propose three localized topological control schemes to improve the network connectivity around critical points to lessen their importance and improve the network resilience. Numerical studies to evaluate the proposed schemes under node and link failure network conditions are presented.**

***Index Terms*—Connectivity, Mobile Ad-Hoc Networks, Resilience, Topology control**

I. INTRODUCTION

Mobile ad hoc networks (MANETs) are expected to become a major component of communication technology in the future. MANETs are comprised of mobile nodes which can dynamically self organize into arbitrary temporary "ad hoc" topologies, allowing users and devices to seamlessly network without a pre-existing communication infrastructure. A variety of applications have been envisioned for MANETs, such as military battlefield communications, disaster recovery emergency communication services, ad hoc extensions to cellular networks to extend coverage, conference networks, etc. In a MANET, the mobile nodes must cooperate to dynamically establish routes using wireless links and routes may involve multiple hops with each node acting as a router. Since the mobile network nodes can move arbitrarily, the network topology is expected to change often and unpredictably. Hence, ad-hoc networks require highly adaptive protocols and efficient failure recovery strategies to deal with the frequent topology changes. MANETs also inherit the traditional problems of wireless communications and networking (e.g., broadcast communication channels, energy constraints, links that are poor quality in comparison to wired links, hidden terminal problems, etc.), which when combined with the unique mobility and lack of infrastructure features make their design and development challenging [1].

A fundamental problem in MANETs and unstructured sensor networks is maintaining connectivity. A network is connected if all nodes have a communication route (typically multi-hop) to each other. Maintaining connectivity is a challenge due to the unstructured nature of the network topology and the frequent occurrence of link and node failures due to interference, mobility, radio channel effects and battery limitations [1].



(a) Bridge link DE and Critical nodes D, E (b) Critical node D

Fig. 1. Example of Critical Connectivity Points

Here we look at how one can develop techniques to improve the resilience of a given MANET without increasing the node density. We assume the network is connected and then determine the *weak or critical points* in the network with the end goal of strengthening the network connectivity around the critical points, thereby improving the network survivability. For MANETs and sensor networks the critical points of the network can be defined as those links and nodes whose failure results in the topology being partitioned into two or more component networks. For example, the link D-E in Figure 1 (a) is a critical point, since the failure of link D-E partitions the network into 2 component clusters. In the literature, links whose failure results in partition of the network are termed "bridge links". Similarly an "articulation" or critical node is defined as a node that partitions the network due to its failure. In Figure 1 (b), node D is a critical node because the network is partitioned if node D fails.

In order to prevent failures from partitioning a network, many researchers have recommended the network topology be *k-connected,* that is, the network should have *k disjoint* routes between each node pair. These *k* routes may be link (i.e., edge) disjoint or node disjoint. Recently several papers have looked at determining conditions under which connectivity [3,4] and *k-node connectivity* [5-7] can be inferred probabilistically or assured asymptotically in MANETs. The focus has largely been on what combination of *node density* and *power range* are required to provide *k*-node connectivity in a specific deployment scenario for a homogenous network. Bettstetter [5]

978-1-4244-5048-0/09/$26.00©2009 IEEE

considered a uniform distribution of homogeneous nodes in a rectangular deployment area and derived a relationship between the minimum transmission range and the probabilistic behavior of the minimum node degree (i.e., number of neighbor nodes). Ling and Tian [6] extend the work of Bettsetter to incorporate deployment area border effects on the range required to provide k-connectivity. They develop and upper and lower bounds on the probability the network is k connected as a function of the transmission range, node density and the perimeter of the bounded deployment space. In [7], the critical transmission power is proposed to maintain *k-connectivity* based on Bettsetter [5]. While results such as [5-7] are of theoretical interest, they require a very high node density to ensure *k-connectivity* which would lead to interference and low throughput in real networks. Furthermore, we show in [15], that the asymptotic nature of the current results make them quite inaccurate in sparse to medium density MANETs.

It is worth noting that ensuring every network node has k neighbors is a necessary condition for *k-connectivity* but not a sufficient condition. This is because the network graph may have critical connectivity points that partition the network when they fail. Only recently has literature begun to appear attempting to define and find the critical points of MANET and sensor networks. Goyal and Caffery [8] use a Depth First Search algorithm (DFS) to find the critical links that may partition the network (critical nodes are not considered in this work). Milic and Malek [9] introduce the Distributed Breadth First Search Algorithm (dBFS) for critical link and node detection using a distributed computation and information technique. Jorgic, et. al., [10] propose heuristics for critical node and link detection utilizing local topology and location information. As noted in [10], locally detected critical points may not be global critical points due to the existence of alternate routes outside the local topology information.

Here we propose a novel algorithm for identification of critical points in a network. The approach is based on results from algebraic graph theory. Unlike existing techniques our algorithm can be used to find multiple failure cases that can partition the network. Additionally the proposed algorithm needs only a simple modification for dealing with limited topology information. Using the critical nodes identified by our algorithm, we propose three techniques to improve the survivability of the network. The basic idea is adjusting the transmission power of individual neighbor nodes of a critical point in order to create additional backup links between the nodes. The rest of the paper is organized as follows. In Section II, the proposed critical point detection algorithm is presented, along with numerical results utilizing the proposed algorithm to study the effects of network density and the effects of limited topological information. We propose three new resilient techniques to strengthen the network around critical points in Section III. Simulation results illustrating the effectiveness and tradeoffs of the resilience schemes are given in Section IV. Lastly, we present our conclusions in Section V.

II. Critical Point Identification

Consider an arbitrary MANET topology of N nodes. Let G be the graph of the topology $G = (V, E)$, where V is the set of nodes, $\{\upsilon_1, \upsilon_2, \upsilon_3,\ldots, \upsilon_N\}$ and E is the set of links. Table 1 summarizes the notation we adopt. Let $A(t)$ denote the N x N adjacency matrix at time t

TABLE 1
Notations used

Notations	
G	Network Graph
V	Nodes in G, $V(G) = \{\upsilon_1, \upsilon_2, \upsilon_3,\ldots, \upsilon_N\}$
E	Links in the network G, $E(G) = \{a_{11}, a_{12}, \ldots, a_{NN}\}$
N	Number of nodes in the network
A	Adjacency matrix of the network graph G
a_{ij}	Link between node i and j ($a_{ij} = 1$ if direct link exists, otherwise $a_{ij} = 0$)
B_i	Set of neighbor nodes of node i. $B_i = \{b_1, b_2, \ldots, b_{di}\}$
d_i	Node degree of node i
D	Diagonal matrix of node degrees
r_i	Transmission range of node i
$dst(A,B)$	Distance between node A and B
N_{CL}	Number of connected components in the network
C_{ij}	Distance based link cost
P_{nf}	Probability of node failure
P_{lf}	Probability if link failure

$$A(t) = \begin{bmatrix} a_{11}(t) & a_{12}(t) & \cdots & a_{1N}(t) \\ a_{21}(t) & a_{22}(t) & \cdots & a_{2N}(t) \\ \vdots & \vdots & & \vdots \\ a_{N1}(t) & a_{N1}(t) & \cdots & a_{NN}(t) \end{bmatrix} \quad (1)$$

where

$$a_{ij}(t) = \begin{cases} 1, & if\ node\ \ i\ and\ \ j\ are\ connected \\ 0, & otherwise \end{cases}.$$

The link connectivity $a_{ij}(t)$ between two nodes depends on their radio range and can be determined by nodes locally through the exchange of ``Hello" packets. The set of nodes B_i that node i has a direct links with are called the neighbor nodes. Let $d_i(t)$ denote the degree of node $i \in N$ at time t (i.e., $d_i(t)$ equals the number of neighbor nodes of i). Note, that the nodal degree $d_i(t)$ can be determined from the adjacency matrix $A(t)$ by summing up the elements of the i^{th} row or column. We define D as the diagonal matrix consisting of the degree of each node (i.e., $D(t) = diag(d_i(t))$).

$$D(t) = \begin{bmatrix} d_1(t) & 0 & \cdots & 0 \\ 0 & d_2(t) & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & d_N(t) \end{bmatrix} \quad (2)$$

Given the network adjacency matrix $A(t)$ we seek to determine the critical links and nodes in the network. We assume the all links are bidirectional (i.e., $a_{ij}(t) = 1 \rightarrow a_{ji}(t) = 1$). The Laplacian matrix $L(t)$ of a graph is defined in terms of the adjacency matrix $A(t)$ and nodal degree matrix $D(t)$ as

$$L(t) = D(t) - A(t) \quad (3)$$

The eigenvalues $\lambda_1, \lambda_2, \ldots \lambda_N$ of $L(t)$ form what is called the *Laplacian spectrum* of the graph. We order the eigenvalues from smallest to largest and re-label them as $\omega_1, \omega_2, \ldots \omega_N$ (i.e., $\omega_1 = \min\{\lambda_1, \lambda_2, \ldots \lambda_N\}$, $\ldots \omega_N = \max\{\lambda_1, \lambda_2, \ldots \lambda_N\}$). In the algebraic graph theory literature [5], it has been shown that zero is always an eigenvalue of the graph (i.e., $\omega_1 = 0$), and the next smallest eigenvalue ω_2 is known as the *algebraic connectivity* of the graph. If the algebraic connectivity is zero (i.e., $\omega_2 = 0$) then the network is partitioned. In fact, the *number* of zero eigenvalues [5] is equal to the number of connected components of the network N_{CL}. The network is connected (i.e., 1-connectivity) if the number of network components is 1 (i.e., $N_{CL} = 1$), otherwise, the network is partitioned.

We develop our algorithm for critical point identification around testing the multiplicity of the zero eigenvalue. The basic idea is to test a possible critical point by removing it from the network and then forming the Laplacian matrix for the remaining graph and testing for connectivity via computing the multiplicity of the zero eigenvalue. This procedure is repeated for each possible critical point (link or node) or groups of critical points (multiple links or nodes or combination of links and nodes) in the network. Let T denote a set of points (i.e., links, nodes or combinations of the two) in the network to be tested for possible partition of the network. The critical point detection procedure is given in algorithm form below.

Critical Point Test Algorithm

Step 1 : Test point $i \in T$ is chosen to check its critical status.

Step 2 : Eliminate test point i from the adjacency matrix A and recompute the nodal degrees in D. Specifically if i is a node then remove row i and column i from A and adjust D, if i is a link then set the appropriate link values in A to zero and adjust the nodal degrees in D.

Step 3 : Compute the eigenvalues of the Laplacian matrix L.

Step 4 : If there exist more than one zero among the Laplacian eigenvalues (i.e., $\omega_2 = 0$) then i is a critical point, otherwise i is not critical and the network is still connected.

Step 5 : Choose the next test point $i \in T$ and go back to step 2.

The algorithm can be implemented at any network node having the adjacency matrix information. As such it is best suited for MANETs implementing proactive routing protocols where topology information is regularly gathered and disseminated to nodes or at the sink node in a sensor network. Also is could be used in MANETs utilizing reactive routing protocols which exchange local connectivity periodically (e.g., AODV [6]).

The time complexity of our algorithm is largely determined by computational time to determine the eigenvalues, since it tests the second smallest eigenvalue to check the connectivity of the network. There are many efficient algorithms for determining eigenvalues which are $O(n^2)$ where n is the the size of the matrix which in our case is the number of nodes. Comparing the other network connectivity testing algorithms such as DFS and BFS they have a time complexity of $O(n + m)$ where m is the number of links. In a sparse network, the number of links m tends to be less than $n(n-1)/2$ and the time complexity, $O(n + m)$, becomes $O(n^2)$ which is same as that of our algorithm. However, our algorithm provides more information such as the number of clusters that the network is partitioned into and the ability to study multiple failure cases.

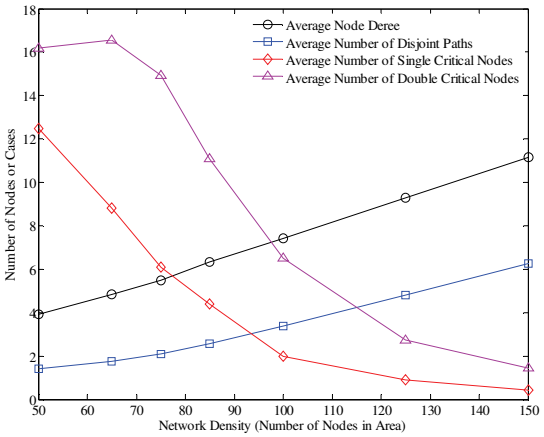


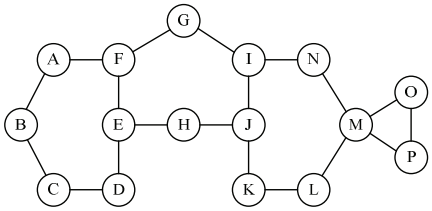
Fig. 2. Average Number of Single Critical nodes, Average Number of Double Critical Nodes, Average Node Degree and Average Number of Disjoint Paths versus the Network Density

A. *Numerical Results*

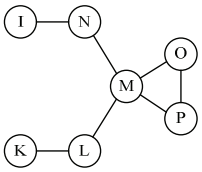
Here, we illustrate the use of our algorithm for detecting critical nodes. The critical point test algorithm was implemented in MATLAB. The behavior of the number of critical nodes is examined for different network densities. In this study, using the ns2 simulator we randomly generate network topologies with different number of nodes (50, 65, 75, 85, 100, 125, 150) in a $1500 \times 1500m^2$ network area. The nodes are randomly and independently distributed in the network area with the (x, y) coordinates determined according to two independent uniform [0-1500] random variables. All nodes are identical and have a 250m transmission range. For each node density, we randomly generate topologies until 100 connected topologies are found. For each network topology we compute the metrics: number of single critical nodes and number of double critical nodes (i.e., any combination of two node failures that partitions the network) in the network. These metrics are then averaged over the 100 topologies for each network density and plotted in Figure 2. From the figure one can see that the average number of critical nodes decreases with increasing network density and the number of double critical nodes is larger than the number of single critical nodes. Note, that the sparser the network, the more likely are critical points. Also, observe that the average number of disjoint paths and average node degree increase with increasing network density but do not match (i.e., average node degree is not a direct proxy for average number of disjoint paths) in part due to the existence of critical nodes.

B. *Critical Point detection using H-hop information*

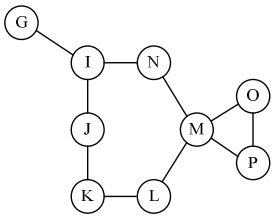
Note, that we can easily adapt the proposed critical point test algorithm to utilize only local topological information as in [10]. Specifically, one uses the algorithm with the *sub-graph* topological adjacency information formed from the *H-hop* neighbors around the testpoint. For example, consider the 14



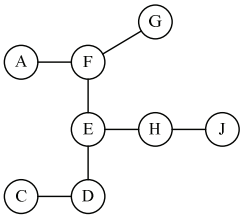
(a) 14 node network



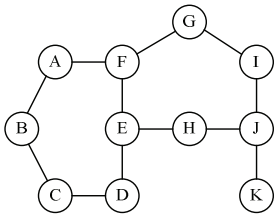
(b) 2 hop local network of Node M



(c) 3 hop local network of Node M



(d) 2 hop local network of Node E



(e) 3 hop local network of Node E

Fig. 3. *H*-hop sub-networks at node M and E (*H* = 2, 3)

node network topology given in Figure 3(a). Further consider the problem of testing whether node M or node E is critical or not using H-hop local information only. Figure 3(b) and (c) show the *2-hop* and *3-hop* connectivity sub-networks of node M, respectively. Similarly, Figure 3(d) and (e) show the *2-hop* and *3-hop* sub-networks of node E, respectively. In order to apply the critical point test algorithm, one simply treats the *H-hop* sub-network as the network topology and runs through the algorithm with the testpoint of node M or E. Note that working with either the *2-hop* or *3-hop* sub-network of Figure 3(b) or (c), the algorithm will indicate that Node M is a *local critical node* when in fact node M is a global critical node. Meanwhile, local testing of node E results in finding out that it is a *local critical node* in *2-hop* sub-network while it is not in *3-hop* sub-network. In global case node E is *not* a critical node as alternate routes exist via node F. These results indicate that the false detection using *H*-hop sub-network depends on *H* value. In general, for *any localized test*, if only local *H-hop* connectivity information is known, *false positives on critical nodes or links* will occur when the alternate routes are longer than the *H-hop* limit. It is worth noting that hop count limits on routes are often used in networks for performance reasons (e.g., end-to-end delay bounds). Also, we observe that the set of global critical points will be contained in the set of all local critical points identified by the algorithm with *H-hop* information. Hence, unlike the algorithms in [10], the critical test point algorithm will have a 100% global critical point detection rate.

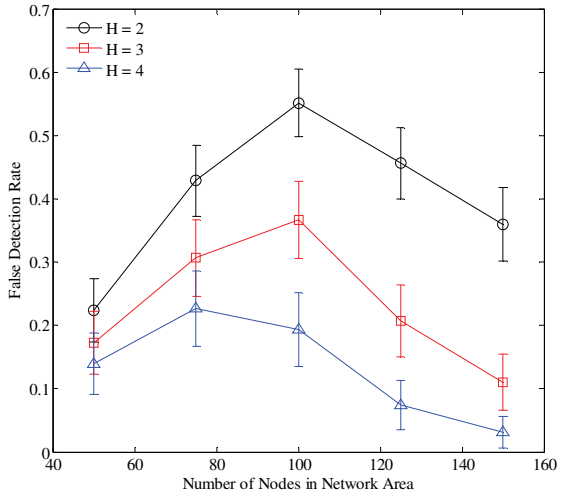


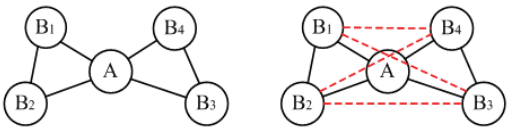
Fig. 4. Single critical node False Detection rate using *H-hop* sub-networks

To illustrate the effects of limited information on critical point detection, we conducted numerical experiments using our critical point test algorithm at each node with the *H-hop* adjacency matrices. We test 100 topologies that are used in above numerical study for each network density (50, 75, 100, 125, 150 nodes in a *1500* x *1500* m^2 network area) with same network and node conditions. For each node in every topology we form the *H-hop* adjacency matrix for *H* = {2, 3, 4} and execute the critical point detection algorithm to test for critical nodes. The false detection ratio is calculated by dividing the number of falsely detected critical nodes by the total number of detected nodes. As shown in Figure 4, the false detection ratio is always lower with larger *H* value since the larger *H* value means that the *H-hop* local information is getting closer to the global network topology.

III. K-CONNECTIVITY VIA CRITICAL NODE MANAGEMENT

As noted earlier if the network topology has k-node connectivity it protects the network from any combination of single, double, triple, … up to (*k-1*) node failures. Thus, if there are no critical nodes up to and including the (*k-1)* multiple node combination case, the network is *k-connected*. Here we propose a critical node management approach to providing *k-connectivity*. Specifically one uses the algorithm of section II to identify critical connectivity points in multiples from 1 to (*k-1*), one then uses topology control via transmission power adjustment at nodes in order to reduce the number of critical nodes to zero, resulting in a network that is *k*-connected. In the remainder of the paper we concentrate on k = 2 connectivity and providing techniques to eliminate critical nodes only.

We present three localized topology modification schemes to increase the resilience of the network by eliminating a critical node namely: (1) Local Full Mesh (LFM), (2) Least Number of Links with Random Selection (LNLRS), and (3) Least Number of Links with Least Cost (LNLLC). The first technique adds all possible additional links to create a fully meshed network around the critical node, while the other two techniques



(a) Critical node A (b) LFM around node A

Fig. 5. Local Full Mesh scheme around critical node A

establish the minimum number of additional links between pairs of neighbor nodes of the critical node to make the node in question no longer critical. All schemes only need the connectivity information between neighbor nodes (i.e., 2-hop). We discuss each in turn below for the single critical node case. An assumption in each case is that nodes have enough power to establish the required new links and for now we ignore interference issues and maximum power limitations.

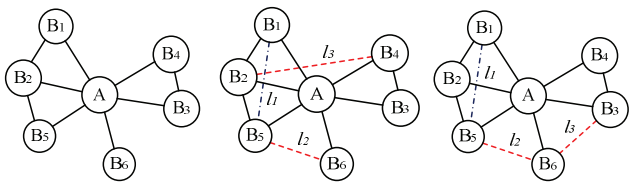
A. *Local Full Mesh (LFM) Scheme*

The Local Full Mesh (LFM) scheme creates a fully meshed local network around a critical node. This scheme simply adjusts the transmission power of all neighbor nodes until all pairs of neighbor nodes have a direct link between each other. For example, in Figure 5(a), the 5 node local network has one critical node at node A. Node A whose node degree is 4 (i.e., $d_A = 4$) has 4 neighbor nodes (i.e., $B_A = \{B_1, B_2, B_3, B_4\}$). Using the Local Full Mesh (LFM) scheme, all nodes in pairs of neighbor nodes who do not have direct link (i.e., B_1B_3, B_1B_4, B_2B_3, B_2B_4) increase their transmission power until they have a direct link to all other nodes. Thus a fully meshed network is established around the critical node as shown in Figure 5(b); the dotted line represents the new links established by adjusting transmission power of each neighbor nodes. Then, the network does not fail due to failure of node A.

B. *Least Number of Link (LNL) Schemes*

The Least Number of Link (LNL) schemes create the least number of backup link(s) among pair of neighbor nodes of the critical node for the node in question to no longer be critical. The LNL algorithms first gathers 2-hop local network connectivity information around a critical node and computes how many clusters the local network partitions into when the critical node fails. For example, in Figure 6(a), node A is a critical node and the number of clusters when node A fails is three ($N_{CL} = 3$). For the example in Figure 6(a), at least two additional links are required to relax the single point of failure problem. Here we propose two schemes for producing the minimum number of required links.

The first scheme is the Least Number of Links with Random Selection (LNLRS) technique. LNLRS randomly chooses a pair of unconnected neighbor nodes and checks if this additional connectivity reduces the number of clusters in 2-hop local network when the critical node is removed. If it does, then chosen pair of nodes is selected otherwise the link is discarded. This process continues until the number of clusters becomes one without the considered critical node. For example in Figure 6(b) LNLRS may chooses a pair of neighbor nodes B_1 and B_5 and creates the link l_1 in Figure 6(b). Yet, the number of component clusters is still 3 ($N_{CL} = 3$) and this link is discarded. Next LNLRS randomly selects nodes B_5 and B_6 and addition of the link l_2 between them reduces, $N_{CL} = 2$. Hence, this link is selected to become part of the topology. Since $N_{CL} \neq 1$, LNLRS randomly chooses another untested pair of nodes (e.g., B_2 and B_4) adds a connecting link l_3 and repeats the process. Incorporation of l_3 in to the topology results in $N_{CL} = 1$ and it is added to the local topology and the algorithm terminates as single point of failure having been removed. LNLRS has the advantage of needing little knowledge of the topology, however due to the random selection, energy expensive cost links may be selected (e.g., B_1B_6 and B_2B_3) in the solution.



(a) Local Network at critical node A (b) LNLRS example (c) LNLLC example

Fig. 6. Additional link selection in LNL schemes

The Least Number of Links with Least Cost (LNLLC) selects the least cost links. Let C_{ij} denote the link cost between pair of nodes i and j, here we set the link cost equal to the distance $dst(i,j)$ between nodes i and j since the power required is a function of the distance. Thus, the larger the distance between two nodes the more expensive the link cost. Therefore, the LNLLC method selects the shortest distance pair of neighbor nodes among disjoint component clusters that do not have a direct link. For example links B_5B_6 and B_3B_6 in Figure 6(c). The least cost links can be computed with acquisition of node position. The node position can be obtained by Global Positioning System (GPS) or localization techniques [13, 14]. While in this paper, the distance between two nodes in used for the cost measure in least cost link selection of LNLLC, other cost metrics could be used such as, delay, SNR, BER, and ETX.

IV. Numerical Study

We evaluate the effectiveness of our critical node management schemes using simulation. Using the ns2 simulator, we generate random topologies with different number of nodes (i.e., *50*, *100*, and *150*) in a network area of *1500* × *1500* m^2. The nodes are independently distributed according to a uniform [0-1500] random variable in the network area. For each network density, we generate 30 connected random topologies where every pair of nodes has at least one route (i.e., they are k = 1 or greater connected) and at least one critical node in the topology. A free space propagation model is used in the simulation. We assume all nodes are identical and have a capability of adjusting transmission power with initial power whose transmission range of 250m. We developed an extension to ns2 to implement our proposed critical node management schemes (LFM, LNLRS, LNLLC). For comparison we implement a well known Minimum Node Degree (MND) scheme based on increasing the node power until every node has k neighbors [3,4].

TABLE 2
CONNECTIVITY PERCENTAGES OVER 30 TOPOLOGIES FOR $K = 2$

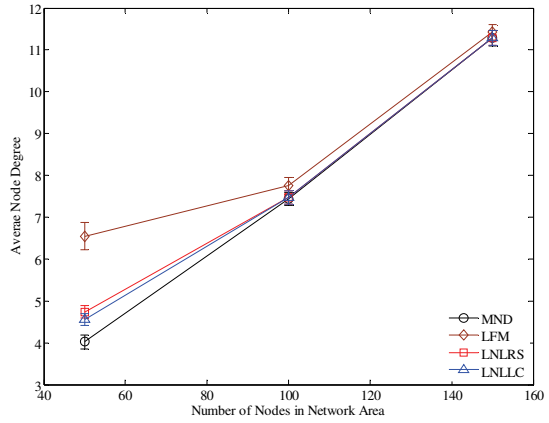
N	No Protection	MND	LFM	LNLRS	LNLLC
50	0%	6.67%	100%	100%	100%
100	0%	36.67%	100%	100%	100%
150	0%	43.33%	100%	100%	100%

Note: Rate is the ratio of number of k = 2 connected topologies to total number of 30 topologies; Minimum Node Degree (i.e., $d_{min} = 2$)

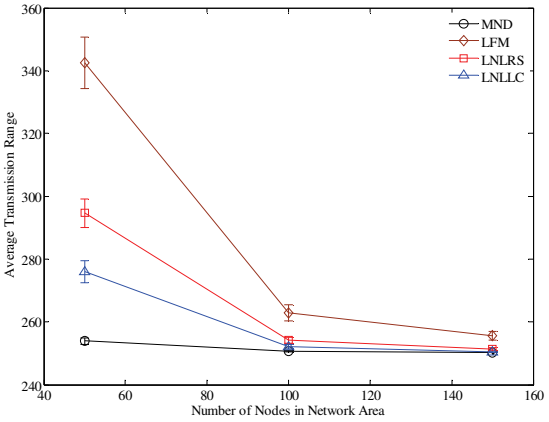
First we examine the effectivness of the proposed schemes in providing k=*2* connectivity for the entire network. Each topology possesses at least one single critical node (i.e., 1-connected). Table 2 shows the percentages of 2-connected networks for each of the schemes for network densities of 50, 100 and 150 nodes. The No Protection scheme corresponds to the original unmodified topology. In MND, the power of every single node is adjusted until minimum node degree requirement (i.e., $d_{min} = 2$) is met. The proposed schemes LFM, LNLRS, and LNLLC, are applied only to single critical nodes to achieve 2-connected network. One can see that the effectiveness of the MND approach varies with the node density, whereas the proposed LFM, LNLRS and LNLLC schemes always result in a 2-connected network. However, there are some tradeoffs between LFM, LNLRS and LNLLC. We consider the average node degree and average transmission range as the tradeoffs and they are shown along with 95% confidence intervals (CI) in Figure 7(a) and (b), respectively. The average node degree provides a metric of connectivity and interference. In the sparse network case (i.e., 50 nodes network), LFM has the highest average node degree (i.e., MND 4.02, LNLLC 4.55, LNLRS 4.75, LFM 6.55). When the network is denser, the average node degrees of the proposed resilient schemes and MND are closer with parts of the confidence intervals overlapping.

The average transmission range can be related to the average energy consumption of the network since increasing the transmission range is achieved by increasing the transmission power of the node. As shown in Figure 7(b), LFM requires significantly more energy than the other schemes for the 50 nodes network (i.e., 342.46 m) case since it requires a larger range. LNLRS requires the second largest range followed by LNLLC as expected. As network density increases, LFM still consumes more than others but it is not as significantly large as it is in 50 nodes network. Both LNL schemes consume almost about the same energy as MND does but the LNL schemes provide full resilience to any single node failure unlike MND.

To further examine the resilience of the proposed schemes we randomly fail nodes and links in the network and determine the probability the network remains connected. Each node fails according to probability P_{nf} and each link with link failure probability P_{lf}. The probability of node failure was set to result in an average of one node failure for each network density (i.e., $P_{nf} = 1/N$). The link failure rate was varied ($P_{lf} = 0.00, 0.025, 0.05, 0.1$), where $P_{lf} = 0.1$ means that on average 100 x P_{lf} = 10 % of the links fail in the network. For each of the thirty topologies we randomly generate 100 experiments for each P_{nf} and P_{lf} and determine the probability the network is connected.



(a) Average Node Degree at $k = 2$

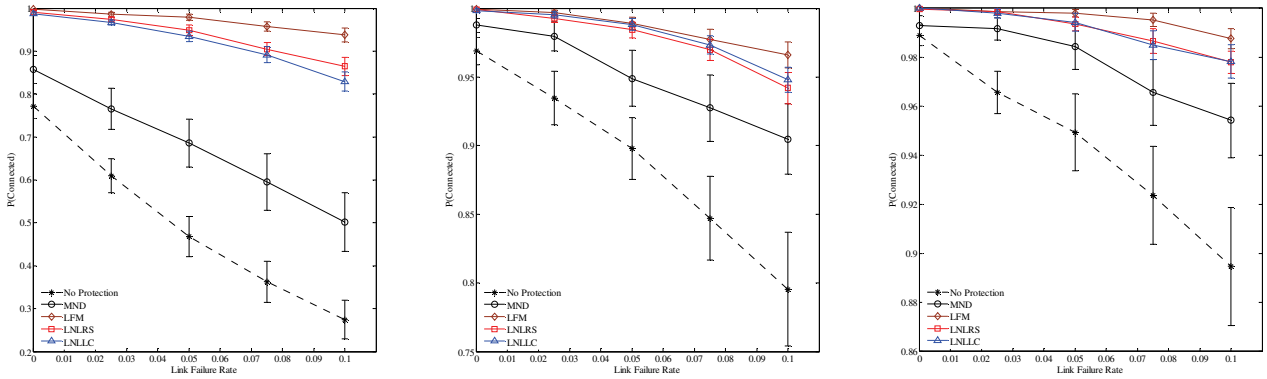


(b) Average Transmission Range at $k = 2$

Fig. 7. Average Node Degree, and Transmission Range at $k = 2$

The probability of the network being connected along with a 95% confidence interval on the estimate is computed and plotted for 50, 100, 150 node networks as shown in Figure 8(a), (b), and (c), respectively. As one would expect LFM improves *P*(*Connected*) the most. For example, for the 50 node network case, the LFM scheme provides a greater than 90% chance the network is connected even with $P_{nf} = 0.02$ and $P_{lf} = 0.1$. As the network density increases, *P*(*Connected*) increases for each scheme. For example, for a 150 node network with link failure of 0.1, *P*(*Connected*) becomes 0.8947 without resilient techniques while LFM improves it up to 0.9877 and MND improves it to 0.9543 for the minimum improvement.

When the three proposed resilient schemes are compared with the Minimum Node Degree (MND) technique, all of them improve the probability of network connectivity more than MND at any network density. Another observation is that *P*(*Connected*) decreases faster with MND as the network is experiencing more severe link failure. For example, in the 50 node network case, when the link failure rate increases from 0 to 0.1, *P*(*Connected*) decreases from 0.858 to 0.502 with MND while it decreases from 0.9877 to 0.8297 with LNLLC (LNLLC has the largest decrease in *P*(*Connected*) among the three resilient schemes).



(a) 50 nodes network with $P_{nf} = 0.02$ (b) 100 nodes network with $P_{nf} = 0.01$ (c) 150 nodes network with $P_{nf} = 0.0067$

Fig. 8. Probability of Network being connected with 95% CI utilizing Minimum Node Degree, Local Full Mesh (LFM), Least Number of Link with Random Selection (LNLRS) and Least Cost (LNLLC) in (a) 50 node with $P_{nf} = 0.02$, (b) 100 node with $P_{nf} = 0.01$, (c) 150 node network with $P_{nf} = 0.0067$

Among the proposed resilient schemes, LFM provides the highest chance to be connected with the given random node and link failure conditions in a sparse network (i.e., 50 nodes). As the network density increases *P*(*Connected*) with the LNL schemes approaches to that with LFM. For example, *P*(*Connected*) with LFM is 0.9377 and with LNLLC is 0.8297 at 50 nodes with $P_{lf} = 0.1$, while it becomes 0.9877 with LFM and 0.9783 with LNLLC at 150 nodes with $P_{lf} = 0.1$. Between the two LNL schemes, LNLRS and LNLLC perform similarly. In a sparse network, LNLRS performs slightly better than LNLLC (i.e., 50 node case) while they have almost the same value of *P*(*Connected*) in a denser network (i.e., 150 node network).

Note that while the LFM scheme has a higher *P*(*Connected*) under node and link failure. However, it consumes more energy and it will be the best scheme in sparse network if the network has unlimited or rechargeable power sources. Otherwise, either LNL scheme is better because they create significantly less node interference and less energy consumption than LFM while being more survivable under node and link failures compared to MND. Between LNL schemes, LNLLC is better than LNLRS since LNLLC performs similar with less energy consumption.

V. CONCLUSIONS

We have proposed a new algorithm to identify the critical connectivity points of a MANET or sensor network topology utilizing results from algebraic graph theory. Unlike the existing algorithms, the proposed technique can test for multiple failure critical points and has a simple implementation using only local information at the expense of false positives in the results. Three resilient schemes that strengthen the critical nodes by utilizing transmission power control are proposed and shown to provide 2-connectivity. The results of a simulation study indicate that the proposed resilient schemes outperform an existing minimum node degree approach in sparse networks. Furthermore, the Least Number of Links with Least Cost scheme is considered the best approach for improving the network resilience will minimizing energy and interference.

REFERENCES

[1] I. Chlamtac, M. Conti, and J. J. Liu, "Mobile ad hoc networking: imperatives and challenges," *Ad Hoc Networks*, vol. 1, no. 1, pp. 13-64, July 2003.

[2] J. Sterbenz, R. Krishnan, R. Hain, A. Jackson, D. Levin, R. Ramanathan, and J. Zhao, ``Survivable Mobile Wireless Networks: Issues, Challenges, and Research Directions,'' *Proc. of ACM Wireless Security Workshop (WiSe'02)*, Sept. 28, 2002, Atlanta, GA.

[3] P. Gupta and P. Kumar, "Critical power for asymptotic connectivity in wireless networks", *Stochastic Analysis, Control, Optimization and Applications*: A Volume in Honor of WH Fleming, 1998. **3**(20): p. 547-566.

[4] Xue, F. and P. Kumar, "The Number of Neighbors Needed for Connectivity of Wireless Networks", *Wireless Networks*, 2004. **10**(2): p. 169-181.

[5] C. Bettstetter, "On the minimum node degree and connectivity of a wireless multihop network," *Proc. 3rd ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, Lausanne, Switzerland, pp. 80—91, June 9-11 2002.

[6] Q. Ling and Z. Tian, ``Minimum Node Degree and k-Connectivity of a Wireless Multihop Network in Bounded Area," *Proceedings of IEEE Globecom*, 2007.

[7] H. Zhang and J. C. Hou, "On the critical total power for asymptotic k-connectivity in wireless networks," *IEEE/ACM Transactions on Networking*, April, 2008.

[8] D. Goyal and J.J. Caffery, "Partitioning Avoidance in Mobile Ad Hoc Networks Using Network Survivability Concepts," *Proceedings IEEE ISCC'02*, Sicily, 2002.

[9] B. Milic and M. Malek, "Adaptation of the Breadth First Search Algorithm for Cut-edge Detection in Wireless Multihop Networks," *Proc. 10th ACM Symposium on Modeling, analysis, and simulation of wireless and mobile systems (MSWiM)*, Chania, Crete Island, Greece, Oct., 2007.

[10] M. Jorgic , N. Goel, K. Kalaichevan , A. Nayak , I. Stojmenovic,`` Localized Detection of k-Connectivity in Wireless Ad Hoc, Actuator and Sensor Networks," *Proc. of 16th IEEE International Conference on Computer Communications and Networks (ICCCN)*, Aug., 2007.

[11] C. Savarese, J. M. Rabaey, and J. Beutel, "Location in Distributed Ad-Hoc Wireless Sensor Networks," *Proc. IEEE Int'l Conf. Acoustics, Speech, and Sgnal Processing*, Vol. 4, pp. 2037-2040, May 2001.

[12] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. Abdelzaher, "Range-Free Localization Schemes for Large Scale Sensor Networks," *Proc Ninth ACM Int'l Conf. MOBICOM*, pp. 81-95, Sept. 2003.

[13] C. Godsil and G. Royle, Algebraic Graph Theory, Springer-Verlag, 2001.

[14] C. Perkins, "Ad hoc on demand distance vector (AODV) routing," IETF-Draft, draft-ietf-manet-aodv-04.txt, Oct., 1999.

[15] T.-H. Kim, D. Tipper and P. Krishnamurthy, ``Connectivity and Critical Point Behavior in Mobile Ad Hoc and Sensor Networks," *to appear Proceedings of IEEE ISCC'09, July, 2009.*