CROSS-LAYER RESILIENCE BASED ON CRITICAL POINTS IN MANETS

by

Tae-Hoon Kim

B.S., Mechanical Engineering, Oklahoma State University, 1997

M.S., Telecommunications Program, University of Colorado at Boulder, 2002

Submitted to the Graduate Faculty of

Networking and Telecommunications Program

in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

University of Pittsburgh

2010

UNIVERSITY OF PITTSBURGH

School of Information Science

This dissertation was presented

by

Tae-Hoon Kim

It was defended on

December 1, 2010

and approved by

David Tipper, Associate Professor, Networking and Telecommunications Program

Prashant Krishnamurthy, Associate Professor, Networking and Telecommunications Program

Richard Thompson, Professor, Networking and Telecommunications Program

Martin Weiss, Professor, Networking and Telecommunications Program

A. Lee Swindlehurst, Professor, Department of Electrical Engineering & Computer Science at University of California at Irvine

Dissertation Advisor: David Tipper, Associate Professor, Networking and Telecommunications Program

CROSS-LAYER RESILIENCE BASED ON CRITICAL POINTS IN MANETS

Tae-Hoon Kim, PhD

University of Pittsburgh, 2010

A fundamental problem in mobile ad hoc 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. In order to effectively deploy techniques to improve the resilience of sensor and mobile ad hoc networks against failures or attacks one must be able to identify all the weak points of a network topology. Here we define the weak or critical points of the topology as those links and nodes whose failure results in partitioning of the network. In this dissertation, we propose a set of algorithms to identify the critical points of a network topology. Utilizing these algorithms we study the behavior of critical points and the effect of using only local information in identifying global critical points. Then, we propose both local and global based resilient techniques that can improve the wireless network connectivity around critical points to lessen their importance and improve the network resilience. Next we extend the work to examine the network connectivity for heterogeneous wireless networks that can be result due to factors such as variations in transmission power and signal propagation environments and propose an algorithm to identify the connectivity of the network. We also propose two schemes for constructing additional links to enhance the connectivity of the network and evaluate the network performance of when a random interference factor occurs. Lastly, we implement our resilience techniques to improve the performance.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

PREFACE

This dissertation is based on my recent research of the Cross-layer Resilience in the MANETs using weak point approach in the Networking and Telecommunications Program in the School of Information Sciences at University of Pittsburgh. While I am completing my Ph.D., I have had many helps that I could not finish without them. For the valuable memories, I want to note with my gratitude.

For the first of all, I want tank to GOD who always be with me, gives me strength, and drove me here for completing my Ph.D. Next, I want to give many thanks to my parents; I could not complete my study without my father, Changsup Kim, who always believes and supports me and my mother, Hyeshin Kim, prays for my family. My wife, Seolhee Yang, is the most precious gift in my life. She devotes herself to caring and supporting me. She also brought me two priceless sons, Joshua and Isaiah. My family has waited for long time with a good patience.

I also give many thanks to my advisor, Dr. David Tipper, who allowed me to join part of his research. He guides my research, supports, and advises me during my study. I also thank to my committee members, Dr. Krishnamurthy, Dr. Thompson, Dr. Weiss, and Dr. Swindlehurst, for their valuable comments on my dissertation.

I also thank my office mates, Korn Vajanapoom, Peera Pacharintanakul, and Sira Akavipat for their kindness. I cannot forget my colleagues, Sung-Min Kim and Jong-Do Park, who gave me good relaxations. Finally, I also want to thank whoever I forget mentioning.

1.0 INTRODUCTION

Mobile ad-hoc networks (MANETs) are expected to become an important part of the communications landscape. 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. 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, asymmetric channels and signal propagation, energy constraints in mobile nodes, links that are poor quality in comparison to wired links, hidden terminal and exposed terminal problems, etc.), which when combined with the unique mobility and lack of infrastructure features make their design and operation challenging

A basic problem in MANETs and unstructured sensor networks is achieving and 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 failures. Several

researches have studied how to make the network more survivable. In this thesis, we are interested in what is an effective way to ensure the network connectivity is robust and services survivable.

Chapter 2 presents the literature review regarding existing techniques for the survivability of mobile ad hoc and sensor network. Several routing protocols to maintain the communication when failure occurs are explained. Another approach is to maintain the network connected in connectivity problem. Lastly, recent approach to locate the weak point of the network in a connected network including topology control is discussed.

In Chapter 3, the connectivity problem is considered and examines some of the existing connectivity results in the literature examined. In particular we evaluate the relationship between minimum node degree and *k*-connectivity. Simulation results indicate that the assumption of the probabilistic relationship being approximately equal does not hold for sparse ad hoc networks.

Chapter 4 introduces the concept of the weak point of the network which is the node or link that has a severe impact on the network when failed. Several metrics such as maximum, minimum node degree, maximum utilizing in primary and backup route, are examined to identify the weak points. Results show none of them are valid due to existence of critical points (i.e., bridge link or articulation node). Two algorithms are proposed to identify the critical nodes and links and critical point behaviors are studied.

Chapter 5 proposes two types of resilient schemes, localized and globalized. Localized resilient scheme uses the knowledge of local subnetwork such as 1-hop neighbor nodes to protect the weak points. Globalized resilient scheme uses the global topological information to find the minimum number of additional links to protect the network from any single point of failure.

Chapter 6 introduces how the network topology changes when heterogeneous wireless network conditions are applied to the at least 1-connected in homogeneous wireless network condition. And two schemes are proposed to improve the network connectivity in heterogeneous wireless network.

Chapter 7 discusses the conclusions and possible future works.

2.0 LITERATURE REVIEW

Chapter 2 reviews the literature on the survivability of mobile ad hoc and sensor networks. Firstly, the general survivability information under wireless environments is presented. Then, survivability techniques in routing layer will be reviewed.

2.1 BACKGROUND

Prior to reviewing the survivability of wireless network, the failures in wireless network must be defined and classified.

2.1.1 Failure Models

In a wireless network, failure can be classified as a physical fault or a software fault [2]. Koroma and *et al.* [2] present three types of wireless LAN physical faults of the end to end communication such as node faults, power faults, and link faults.

Node faults

Node faults occur when an intermediate node, such as a router and switch, is not available [2,3]. In Figure 1 (a), if node C1 fails, then node A1 and B1 are not able to communicate with each other because C1 acts a router between A1 and B1. If the other possible

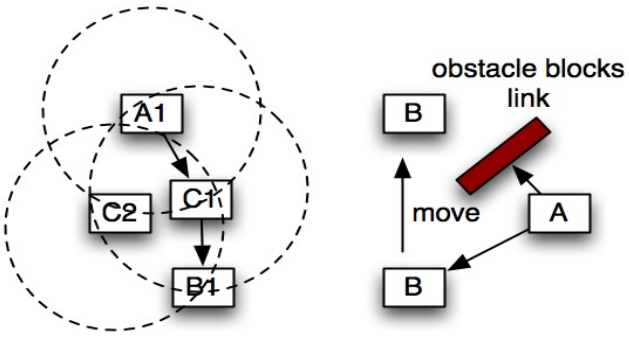
choice for an intermediate router, C2 is unavailable, then there is no way for A1 and B1 to communicate. These faults may be caused by other hardware component failures, software configuration errors or software failures.

Power faults

Power faults are caused due to the limitation of battery life. An intermediate node, which acts as a router, can die when its battery is too low to serve as a router [2,3].

Link faults

Link faults are caused by the wireless network environment, for example, obstacles between communicating nodes and excessive noise as shown in Figure 1 (b). When node B is moving while it is connected to A and obstacles or excessive noise are present, the link between A and B may no longer work until obstacles are removed or excessive noise is reduced.



(a) Node failure (b) Link failure

Figure 1. Example failures in mobile ad hoc network: (a) node failure such as low battery power and (b) link failure such as node movement

2.1.2 Survivability

The goal of survivability is to recover the network system in order to provide a certain level of Quality of Service whenever failure and/or attack are present [3]. Survivability in wireless

networks is different from that in wired networks. However, the survivability techniques for wired networks cannot be directly applied to wireless networks due to their unique characteristics, such as node mobility, wireless channel, power conservation, relatively poor quality links compared to wired network, limited frequency resource, and so forth [4,5]. Tipper, Dahlberg, Shin, and Charnsripinyo [5] classified strategies to improve network survivability into three categories, namely: (1) prevention, (2) network design and capacity allocation, and (3) traffic management and restoration. Prevention is mainly focused on improving component and system reliability, such as fault-tolerant hardware architectures, backup batteries, and so forth [5]. Network design and capacity allocation techniques provision more network links in order to provide sufficient diversity and capacity in the network to reduce the impact due to loss of a link or node [5]. Traffic management and restoration is used for minimizing the impact of a failure by redirecting and restoring the load on alternate routes [5].

However, the strategies above are for wireless access networks (cellular networks) [4,5]. Ad hoc network does not rely on infrastructure while cellular networks rely on infrastructure. Due to this characteristic, in [6], they mentioned that survivability focus on infrastructure is not quite suitable for ad hoc network. Therefore, the survivability in ad hoc network is distinguished from cellular network and it has to adapt cellular network survivability schemes as appropriate. Sterbenz *et. al*. provided three major thrusts that might help to increase the survivability of MANETs [6]: (1) establishing and maintaining network connectivity, (2) expectation of challenging environment such as common occurrence of weak and episodic communications, communication should be possible, and application should adapt to this, and (3) exploiting technology to achieve better survivability such as adaptive protocols and satellite as a backup communication.

In survivability analysis, system performance has to be analyzed for following three periods; transient period right after a failure, steady state failure period, and following failure recovery [4,7]. Survivability performance of MANETs is not easy to measure due to the characteristics of MANETs. However, it is challengeable study by adapting and modifying existing survivability measures used in wired network. Chen, Garg, and Trivedi [3] introduced the quantitative approach to evaluate the system survivability performance by defining the system survivability as a composite measure of the failure duration and the failure impact on the system. By this definition, they proposed the measure of the excess packet loss due to failures using the system failure duration and the packet losses during each failure.

2.1.3 Graph Theory

A graph G has the sets of vertex and edges [43]. As the network is consisted of sets of nodes and links, the graph theory is used to study the network. In graph theory, the graph is represented by adjacent matrix. In the graph, there is no loop, which indicates that no link returns back to its origin. Due to this characteristic, diagonal entries in adjacent matrix are all zeros. All other entries represent the direct connectivity between pairs of nodes. Consider an arbitrary grapy with N vertex with V edges. The adjacent matrix is shown in equation (1).

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

Each entry a_{ij} indicates the direct link between nodes i and j. If the direct link exists between node i and j, a_{ij} is 1. Otherwise, $a_{ij} = 0$. If the link is bi-directional (i.e., $a_{ij} = a_{ji}$), the adjacent matrix is symmetric. Otherwise, it is asymmetric, which has directed link.

The connectivity of the graph can be computed by using eigenvalue of the Laplacian matrix, which also known as algebraic connectivity. Algebraic connectivity is a second smallest eigenvalue of the Laplacian matrix, which can be computed by subtracting adjacent matrix from diagonal matrix. The diagonal matrix is all zeros but diagonal entries, which is the node degree of the corresponding node. It is also known that the larger value of the algebraic connectivity indicates the better connectivity of the network. In other words, the smaller value of the algebraic connectivity indicates that the graph has the more chance to be disconnected. The graph can be disconnected by removing vertex or edge. They are known as articulation node and bridge link. Articulation node is a node that disconnects if the node is removed. Similarly, bridge link is a link that disconnects the graph when bridge link is removed.

2.2 SURVIVABILITY IN MANETS

Tipper *et. al.* [5] and Sterbenz *et. al*. [6] introduce the three possible starting points of the resilient techniques against network failures in MANNETs. Basically these three points can be defined and distinguish schemes. One is maintaining communication between the source and destination nodes, another is maintaining and adjusting connectivity, and the other is preventing the failure. The first scheme is more focused on maintaining the end to end communication being possible for any reason. The second scheme tries to maintain the connectivity in order for the network to be connected in any failure condition. In other words, the former heals the network

based on each prescription when failure occurs while the later makes the network immune to the failures. The last one is to prevent the network failure.

2.2.1 Maintaining Communication

In this section, we review several techniques that increase the survivability by maintaining the communication between pair of nodes. Based on the current existing network connectivity, routing protocols handle the creation and maintenance of the communication path. Specifically, routing protocol finds the path between the source and destination nodes reactively or proactively depending on routing protocol type and tries to rebuild the path when the path is broken for any reason. In a reactive protocol, the source node finds the path upon demand for communication with the destination node, whereas a source node maintains knowledge of the entire network topology in a proactive protocol. In the same manner, routing protocols can be distinguished reactive vs. proactive to the network failures. The other survivable routing scheme is to find the path that is comprised of the most reliable inks. Here, several different methods to find the reliable link are proposed and it is assumed that the more reliable link keeps alive longer.

2.2.1.1 Reactive to failure

Protocols that react to a failure try to find the alternate route to the destination when the failure occurs. Swam Intelligence (SI) techniques here recently been proposed as a method react to failures in MANETs. Swarm Intelligence (SI) was inspired from ant's behavior. Biologically, it was found that ants are able to find shortest paths using the pheromone trail deposited by other ants [8,9]. By taking ant's behavior, SI has been considered to choose the good routes in MANETs [10] and many studies are ongoing in this field. In SI, a source node obtains the route

via backward reactive ant, which called backward learning. Using a learning mechanism, each node learns network status in SI.

The AntNet algorithm [12] utilizes an agent called an ant to discover the route. Each node has its own cumulated history of path from its each neighbor nodes to destination node and computes the availabilities probabilistically. Then each node chooses the best link to the destination node in order to establish the primary path for the communication. Using periodic monitoring, it discovers a new route when a failure is presented. Ad hoc Networking with Swam Intelligence (ANSI) utilizes the same scheme in [11] not in a stochastic but in deterministic approach. However, this protocol finds the new path to the destination node at the last node that loses the path to the destination node due to failure in link to next hop node or next hop node. If it cannot find new route at the intermediate node, the source node re-initiates the route finding.

2.2.1.2 Proactive to failure

Proactive to failure protocols provision against a failure before it occurs. Several researchers have proposed establishing alternate routes for the communication. The basic idea is to determine a set of alternate routes from the source node to the destination node besides the primary route. The alternate route has to be node or link disjoint; otherwise there would be a single point of failure. If two routes share more than one node or link and the corresponding node or link fails, both routes may fail at the same time. In MANETs, link failure may occur more frequently than node failure due to obstacles, fading, and so forth. There are several different ways to establish multiple disjoint routes in end to end communication.

On-Demand Multipath Routing [13], Split Multipath Routing [14], Alternate Path Routing [15], and AODV-BR (Backup Routing) [16] try to establish an alternate path at the route request phase. On-Demand Multipath Routing has two schemes to find backup paths. One is to

find backup paths from the source node to the destination node. The other is to find the backup paths from all intermediate nodes to the destination node. Split Multipath Routing uses the flooding of route request and replies to identify multiple paths. Each node sends duplicate route finding messages coming on different links in order to find the maximum number of alternate routes between the source and destination nodes. In Alternate Path Routing, the source node gathers full topology information and chooses the best path among the found disjoint paths to the destination nodes. AODV-BR (Backup Routing) uses an overhearing method at the route finding phase. Each node overhears the route reply message of neighbor nodes and establishes a local mesh network. Then, when the failure occurs, the local mesh network is searched to provide the other route to the next hop of the failed link or node.

2.2.1.3 Reliability based routing

Four routing schemes are introduced in reliability based routing, Multi-Path Dynamic Source Routing (MP-DSR), Backup Source Routing (BSR), DSR with Stability, and Reliable ad hoc routing [19-22]. These protocols may have a long time period for the route setup because they have to receive Route Request (RREQ) messages as many as possible to find the more possible reliable path.

MP-DSR [19] measures the end-to-end reliabilities all possible paths between the source node and the destination node and select the most reliable path. The path reliability is an accumulation of the link availabilities upon the path using the link availability based on node mobility and defined by McDonald et al. [17] and Jiang et al. [18]. In route discovery phase, the source node sends Route Request (RREQ) messages and each forwarded RREQ message includes the link availabilities that it traveled. The destination node runs the path selection algorithm based on the collected information from RREQ and finds the shorter path among

disjoint paths. However, this does not guarantee the most reliable paths due to accumulation characteristic that there might be a very high cost link while all other links contain very low cost among the intermediate links of the selected path. Another drawback is that it does not guarantee the disjoint path if the network does not provide sufficient connectivity. BSR [20] finds a backup path using the reliability prediction based on link lifetime. BSR determines the primary route using minimal end-to-end delay metric and it establishes the backup route in order to reduce the frequency of route discovery initiation. Then, it selects the route which has long lifetime of the path as a backup route. The lifetime for each link in the network is approximated using an independent and identical exponential random variable [20]. BSR-flooding and BSR-LCS (Lower Cost Search) are introduced for backup route discovery. BSR-flooding forwards RREQ message only when the node is not destination node and time-to-live value is greater than 0 while BSR-LCS forwards RREQ message if it has less cost than existing backup route. This backup route discovery phase should find the backup route that is disjoint to primary route. However, if there does not exist any disjoint route between the source node and the destination node, this technique does not find backup route. In DSR with stability, the lifetime of the link is also selected as a metric. The life time of the path is determined by the minimum link lifetime among that of the intermediate links. The signal strength is chosen to determine the distance between two direct neighboring nodes. Therefore, the route lifetime is determined by finding the minimum inverse value of the distance between pair of direct nodes. They proposed two criteria to select the path. In Maximum/Minimum Signal Strength (MMSS), the source node chooses the path which has maximum value in minimum signal strength field among the received paths from the destination node where the source node chooses the minimum number of hops among the received paths in Minimum number of Hops and Maximum/Minimum Signal Strength

(MHMMSS). Then, it selects the one has maximum value in minimum signal strength field among them. However, MMSS does not consider the end-to-end delay while MHMMS does not guarantee that the maximum lifetime of the path will be selected. Besides, the receiving signal strength is not reliable to measure the distance between pair of direct neighboring nodes.

Some protocols use link quality such as Expected Transmission Count (ETX) [22] or Expected Transmission Time (ETT) [23] in order to find a more reliable path. The techniques discussed above, such as disjoint alternate path, link availability, or link quality, cannot measure the survivability of the ad hoc network because they have no knowledge about the network connectivity.

2.2.2 Connectivity

An essential characteristic in mobile ad hoc and sensor networks is connectivity. Connectivity allows each pair of nodes to communicate. In a wired network, it is possible and relatively easy to make the network more survivable by increasing the connectivity since the connection between nodes is a cable. Thus the network topology can be designed with survivability requirements in mind [49].

Unlike wired networks, connectivity in ad hoc networks is established by radio propagation. A node transmits with a certain signal power and the transmitted signal attenuates with distance. Equation (2) shows a typical path loss model where α is the path loss exponent and d is a distance.

$$\Pr \propto d^{-\alpha} \tag{2}$$

An any node receiving the signal at a level greater than a signal strength threshold, a direct connection between two nodes is created. A nature of connectivity in ad hoc network makes it different to design the network for robustness because of the randomness of the network topology.

2.2.2.1 1-Connectivity

One Connectivity (1-connected) means that every pair of nodes in a network has at least one path to each other. Thus, the network with 1-connectivity or a 1-connected network represents that each pair of nodes in the network is connected via at least one path. Gupta and Kumar [24] studied the connectivity issue in MANETs and they propose the transmission range that makes the network connected. If each node has transmission range r that satisfies $\pi r^2 = \sqrt{(log(n) + c(n))/n}$ where n nodes exist in an unit disk area, the network is asymptotically connected when the number of nodes approaches infinity (i.e., $c(n) \rightarrow +\infty$). They utilize a probabilistic approach to study the network connectivity. They compute the probability of 1-connected network by finding the probability that the network is disconnected. The probability of network being disconnected is estimated by computing the probability that each node is independently isolated. Their study implies that the network is at least 1-connected when there are a infinite number of nodes. Practically an infinite number of nodes is not possible in wireless ad hoc or sensor network.

Santi and Blough [25,26] study the connectivity in the network with a finite number of nodes. They try to adjust the number of nodes and transmitting range in order to make the network 1-connected. In order to compute the lower bound on the 1-dimension case, they segment the line of length *l* by the transmission range *r* and check if there is any empty segment. If any empty segment exists, the network is disconnected. In other words, the network is

connected if there is no empty segment. Unfortunately, they do not count the case where two nodes are placed on the end sides of two segments as shown in Figure 2.

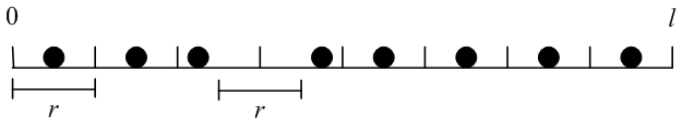


Figure 2. Example of no empty segmentation case but disconnected network

Figure 2 shows the case that there is no empty segment but network is disconnected. In spite of this, they provide the connectivity problem study in 1-dimension such as how many nodes may need to cover certain area and being connected network with a given transmission range. And they extend this problem in 2 and 3 dimensions with the mobility in [27].

Xue and Kumar [28] study the connectivity problem using the number of neighbor nodes (i.e., node degree) for 2 dimensional areas. They argue that if each node has less than $0.074log(n)$ neighbor nodes, then the network is disconnected while it is connected if each node holds more than $5.1774\log(n)$ neighbor nodes. Trap and *k*-filling are used to establish the lower bound, necessary for connectivity. A square area, it is divided into many small square blocks and it is tested for *k*-filling event utilizing a trap concept. The trap is a square of size *d* with 21 small squares of size of the side, *d*/6, in inside of the trap. One block is in the center of the trap and the other 20 blocks are located on the periphery of the trap; small boxes are attached to each side line. A *k*-Filling event is the case that each block in trap holds certain number of near neighbor nodes but no node is presented between center block and all other 20 blocks on the periphery of the trap. This indicates that nodes in center block satisfy the number of neighbor nodes condition but they are disconnected from other. Thus *k*-filling occurs and the network is not connected. For the upper bound, sufficient for connectivity, they divide the area with grid with a side of length 2*r* satisfying $\pi r^2 = (K\log n)/n$ and each corner of the grid holds the small disk with a diameter of

r. If each disk in the area contains no more than certain number of nodes, the network is connected.

Hekmat and Mieghem refine radio model in order to improve the connectivity of the ad hoc network. Connectivity papers shown above are based on geometric random graph melds [29]. They introduce the new radio model is based on the link probability computed in equation (3).

$$p(\hat{r}) = \frac{1}{2}\left[1 - \mathrm{erf}\left(\alpha\frac{\log(\hat{r})}{\xi}\right)\right], \quad \xi \triangleq \alpha/\eta \tag{3}$$

In equation (3), α is calculated as $10/\left(\sqrt{2}\log 10\right)$, $\hat{r}$ is the normalized distance between the transmitter and the receiver, σ is the standard deviation of shadowing, and η is the pathloss exponent where ξ can be calculated as in (3). They try to increase the node degree via increasing the links. They argue that probabilistic link model is more practical than geometric random graph model and it will increase the connectivity. In their theorem, the interference is not counted on link probability. They assume that the interference on the link does not affect the link connectivity but does affect the link capacity. However, interference is one of factor that affects the link loss.

1-Connectivity studied in above papers is simple basic connectivity of the wireless ad hoc and sensor network in order to communicate each other. However, this makes all nodes connected without the failure but does not guarantee its connectivity when a failure occurs.

2.2.2.2 k-Connectivity

The network with *k*-connectivity means that each pair of nodes has *k* disjoint paths in the network. Disjoint paths here means that path does not share any common node or link. In this case, the network is protected from any failures of *k* nodes, links, or combinations of both. This

technique is used in wired networks and is known as (k+1) protection which provides k other predetermined backup paths for each pair of nodes. However, the topology in MANETs is not stable and predetermined backup routes are not always feasible. Thus, there need other ways to make it work.

Recently several papers have looked at determining conditions under which k-node connectivity can be inferred probabilistically or assured asymptotically [30-32]. 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 [30] 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). Furthermore, he notes that the minimum node degree in the network d_{min} can be related to the probability that the network graph G is k-connected (node disjoint) by (4).

$$P(G \text{ is } k-\text{connected}) \leq P(d_{\min} \geq k) \tag{4}$$

He simulates these two probabilities in different transmission ranges over 10000 random topologies. According to the results, those two probabilities are very close when the network as the transmission range increases. When transmission range increases with constant network size, the more links are created and hop count for each pair of nodes decreases. This means that most of pair of nodes tends to have direct link (i.e., 1 hop) when the transmission range is large enough. The more links between nodes increase the probability that the network is k-connected. Then, the probability the minimum node degree is greater than or equal to k can approximate the probability that the network is k-connected when the transmission range is large as shown in

equation (5). In terms of network density, a large number of nodes is used in the simulation study (i.e., 500 nodes over 1000x1000 m^2) [30].

Ling and Tian [32] extend the work of Bettsetter [30] to incorporate deployment area border effects on the range required to provide k-connectivity. The border effect on this problem is that the nodes locating around the border area has relatively less node degree and it makes difference between analytical and simulation studies. 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.

Zhang and Hou use the *k* connectivity assumption in [30] and they propose the critical transmission power to maintain *k*-connectivity by lower and upper bound in [31]. They compute the lower bound of critical power to maintain *k* neighbor nodes. In an upper bound, they introduce the concept of strongly *k*-connected by setting the transmission range to have *k* neighbor nodes in each quadrant. With these lower and upper bound conditions, they draw required transmission power to keep *k*-connectivity.

Another proposed method to get a *k*-connected network is given in [33]. Li *el. al.* where they introduce *k*-connectivity using equation from [34,35]. They use the *k* node degree condition for a lower bound to satisfy the *k*-connectivity and derive the upper bound re-computing the equation from [34,35] with a condition of infinite number of nodes *n*. For the fault tolerance, they use topology control. When a node is required to have *d* of neighbor nodes, it divides the communication disk into certain number that is greater than *d*. Then, a node chooses *d* neighbor nodes among the best nodes (closest one) chosen from each division. This tries to increase the number of disjoint paths by choosing more various neighbor nodes' positions. Whoever this is only valid when sufficient number of node is presented in each direction.

A weakness of the current above literature is the assumption of the relationship between *k*-connected network and minimum nodal degree (or in some cases the average node degree). Specifically, many papers use the approximation that the probability the network graph is *k*-connected is equal to the probability that the minimum node degree is greater than or equal to *k*

$$P(G \text{ is } k-\text{connected}) \approx P(d_{\min} \geq k) \tag{5}$$

Furthermore, the simulation results and analysis used to justify (5) in the literature use very high node densities which would lead to interference and low throughput in real networks. 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.

2.2.2.3 Failure Prevention

Failure prevention could increase the survivability in wireless ad hoc and sensor network. In this section, we assume that the network is failed when partition occurs. In this sense, failure prevention can be done utilizing network partition avoidance. Some researchers have focused on network partition avoidance. The main idea is to prevent the network from partitioning by strengthening the detected bridge link, which partitions the network by its failure, until an alternate route is available.

Jorgic, Stojmenovic, Hauspie, and Simplot-Ryl [36] introduce localized detection algorithms of critical nodes and links. Specifically they propose two algorithms: 1) the *top_critical* algorithm uses local topology information of the network and 2) the *pos_critical* algorithm uses location information via GPS. A node gains the local information using *k*-hop

topological knowledge, GPS equipped nodes, or relative coordinate finding by measuring signal strengths or time delay. However, it is not easy to obtain position information using GPS, signal strength measurement, or time delay due to position error, and environments where GPS is unavailable or weak (indoors, dense urban environment). In [37], they introduce local subgraph connectivity detection (LSCD) algorithm for analyzing the gathered local topology and position information. It has two conditions to declare *k*-connected. First, it checks if each of *p*-hop neighbors has at least *k* degree. Second, it checks if subgraph of *p*-hop neighbors is *k*-connected via disjoint paths. In local critical node detection (LCND), they use the algorithm in [36]. Their results shows that it is difficult to detect critical node correctly based on limited local information as false positives are common.

Goyal and Caffery [38] use the Depth First Search (DFS) algorithm to find bridge links. A node initiates DFS and it finds the links that do not have a sub-loop. In DFS, if there are more than 2 nodes to choose as the next search hop, one node is chosen and the rest of neighbor nodes have to wait until the chosen one finishes the searching. If the sub-network belong to chosen node is very large, the rest of nodes have to wait a long time until it finishes [39].

Milic and Malek [39] introduce the Distributed Breadth First Search Algorithm (dBFS) for bridge link and critical node detection. The dBFS algorithm requires little overhead and uses information collected by reactive routing algorithms in route discovery. In a simulation model, they implement the dBFS algorithms using real network information from a testbed. Topologies are changed by failure of nodes with a probability. Based on the obtained real network link quality information, they simulate the network and calculate the accuracy of dBFS in locating bridge links in a heterogeneous network using OLSR (Optimized Link Source Routing) [40].

2.2.3 Topology Control

Topology control changes the network topological information intentionally in order to achieve a specific goal such as maintaining connectivity. This technique generally requires additional node or link and it can be added by several different methods such as adjusting transmission power, node movement control, or placing additional particle.

Several literatures manipulate the transmission power in order to control the network topology in order to maintain or achieve k-connectivity of the network using node degree [30-35]. The main idea of topology control is adjust the transmission power or range so that the network can create additional links. Those additional links can protect the network from any failure of node or link. However, those topology control literatures only consider the node degree. They increase the transmission power in order to maintain certain minimum node degree in the network. At each node, it increases its transmission power until it reaches the minimum required node degree.

Other researchers focused on topological control by control the node movement [55 – 56]. They control the node's movement around the articulation node and bridge link. However, the node movement control is very difficult because its movement may cause other articulations node or bridge links. It will have more chance to create another point like that when it moves the nodes around those points. And its computation time may be very high.

2.3 SUMMARY

The techniques for maintaining communication are categorized into three groups; reactive [10-12], proactive [14-16] to failure and reliable based [18-23]. These techniques try to find a path between source and destination in order to keep them connected for any failure case. However, these methods are not effective if the network is not connected or partitioned. If there is any node that may partition the network due to its failure, those techniques are not able to guarantee all connections. The connectivity techniques try to provide the sufficient connectivity in order to achieve a connected network. Here, the connected network means all pairs of nodes in the network are connected in direct or multi-hops. To estimate the connectivity of a network, the literature utilizes the number of neighbor nodes (i.e., node degree). However, there can still exist weak points that may partition the network if it fails or limit *k*-connectivity. In other words, the number of neighbor nodes may not be sufficient for the network to be connected. Failure prevention techniques focus on finding the weak point of the network. However, little research has appeared so far in this area and it requires further study.

3.0 CONNECTIVITY AND CRITICAL POINT BEHAVIOR IN MOBILE AD HOC AND SENSOR NETWORKS

A well-known approach to increase the resilience of mobile ad hoc networks (MANETs) and unstructured sensor networks is to ensure a network topology where there are at least *k* disjoint routes in the network between each pair of network nodes (usually called *k*-connectivity). Asymptotic analyses of node density requirements for *k*-connectivity have been considered in the literature. In this Chapter, we present the results of a simulation study investigating the relationship between asymptotic results in the literature and *k*-connectivity under varying nodal density and nodal degree. The numerical results illustrate where the asymptotic approximations breakdown and we show that this largely due to the existence of critical connectivity points in the topology. Using a critical point identification algorithm we examine how the number of critical points varies with nodal degree, nodal density and node mobility. In addition, critical point is evaluated its effectiveness on the network caused by failure.

3.1 K-CONNECTIVITY

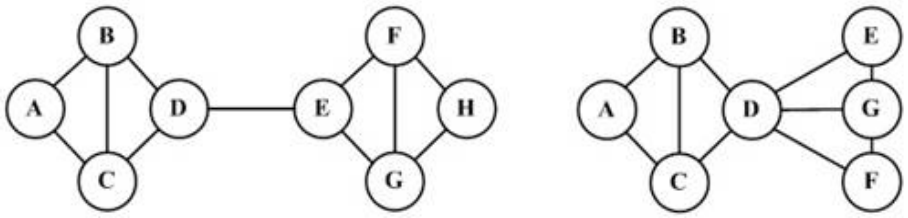
In order to prevent failures from partitioning the network as a whole, many researchers have recommended that the network topology be *k-connected,* that is, the network topology be such that there are at least *k disjoint* routes between each node pair. These *k* routes may be link (i.e.,

edge) disjoint or node disjoint. Since both node and link failures are likely in MANETs and sensor networks, the focus of the research literature has been mostly on node disjoint *k*-connectivity. Ensuring that the network has *k*-node disjoint connectivity results in the network being able to survive the failure of *k*-1 nodes and still remain connected (i.e., at least one route between each node pair).

3.1.1 Node Degree

Node degree is a number of neighbor nodes of a node, where the neighbor node is the directly connected node (i.e., one-hop). In MANETs and sensor network, the nodes in transmission rage are the neighbor nodes in free space. Thus, the number of nodes in transmission rage is the node degree. Recently several papers have looked at determining conditions under which *k-node connectivity* can be inferred probabilistically or assured asymptotically [30-32]. The focus has largely been on what combination of node density and transmission range are required to provide *k*-node disjoint connectivity in a specific deployment scenario for a homogenous network (all nodes are identical). As introduced in Chapter 2, Bettstetter [30] 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). Besides, he remarks that the minimum node degree in the network d_{min} can be related to the probability that the network graph G is *k*-connected (node disjoint). However, the simulation results and analysis used to justify (1) in the literature use very high node densities which would lead to interference and low throughput in real networks. Why such results are theoretically important, a weakness of much of the current literature is the assumption that an equality relationship holds between a *k*-connected network and *k*-minimum

node degree (or in some cases *k*-average node degree) regardless of the scenario. It is worth noting that it is well known in graph theory that ensuring every network node has *k* neighbors is a *necessary condition* for *k*-connectivity but not a *sufficient condition.*



(a) Weak points D and E (b) Weak point D

Figure 3. Examples of weak points that partition the network due to their failure

This is because the network graph may have critical connectivity points. For example, the link D-E in Figure 3 (a) is a critical point. If link D-E fails, the network partitions into 2 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 3 (b), node D is a critical node because the network is partitioned if node D fails.

In the rest of this Chapter, we present the results of a simulation based study on *k*-connectivity and its behavior. We first examine the relation between node degree and its behavior and how good metrics such as the minimum and average node degree are at ensuring *k*-connectivity in Section 3.2. We also discuss and consider the behavior of critical connectivity points and how they are affected by node density and mobility. Section 3.3 studies the performance impact of the failure of critical nodes in terms of the size of the network partitions and the packet loss rate. We conclude the paper in Section 3.4.

3.2 RELATION BETWEEN NODE DEGREE AND CONNECTIVITY

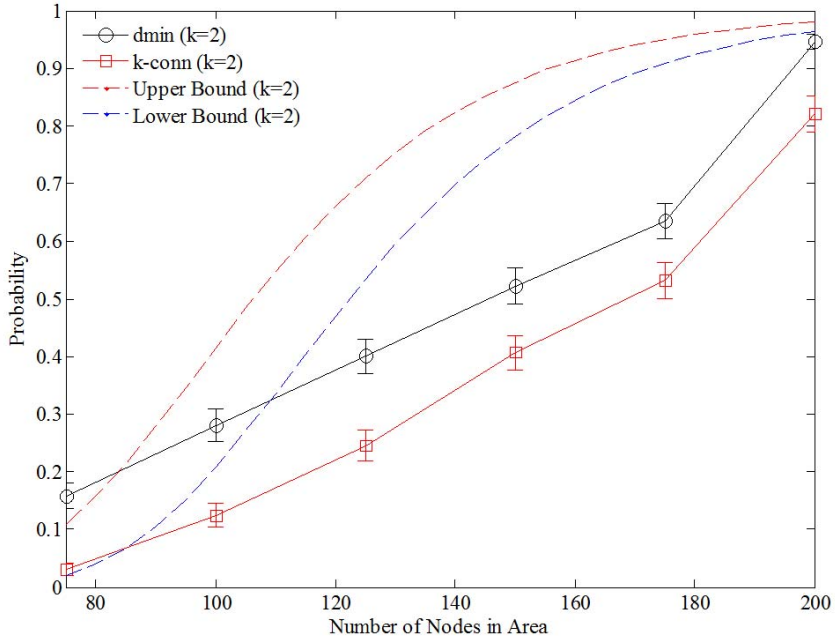
Here we use different simulations to explore the relationship between the node degree and *k*-connectivity of the network. In our simulation models we assume identical nodes with omni-directional antennas and transmission is modeled as a disk of radius *R*. Links between a node and its neighbors will exist only if they fall within the disk (i.e., distance between nodes less than or equal *R*). First, we study the relationship between minimum node degree and *k*-connectivity versus network density.

3.2.1 Minimum Node Degree and *k*-Connectivity

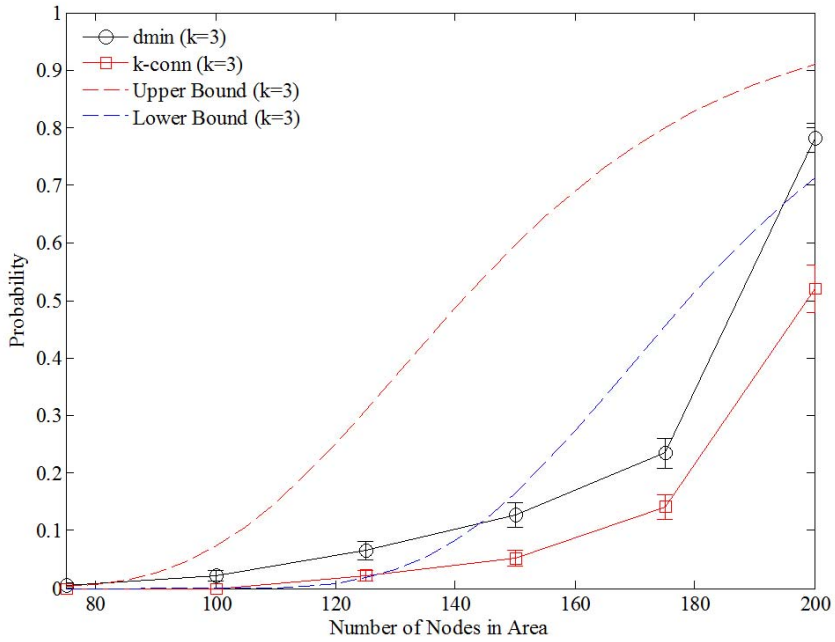
We use the ns2 to generate random topologies with different number of nodes, (75, 100, 125, 150, 175, and 200), in an area of 1500×1500 m^2. Nodes are identical with transmission range of 250m. Once the topologies are randomly generated, a C++ program that we developed was used to evaluate the node degree. The program also implements the k-shortest path algorithm [11], to test the number of *k*-node disjoint routes between each node pair. In each case, we test 1000 randomly generated connected topologies except for the 200 node network case. In the 200 node network density case, 567 random connected network topologies are examined. The probabilities are computed by the fraction of topologies that satisfy the minimum node degree (i.e., $P(d_{min} \geq k)$) or *k*-connectivity (i.e., $P(G \text{ is } k\text{-connected})$).

The simulation results are given in Figure 4 and show the probabilities of having a network with minimum node degree of *k* and *k*-connectivity versus the network density. Note, that the error bars on the results represent 95% confidence intervals. Note, that the probability of *k*-connectivity never reaches the probability of minimum node degree. For example, at network

density of 175 nodes for the $k = 2$ case, $P(d_{min} \geq 2) = 0.635 \pm 0.03$, whereas $P(G \text{ is 2-connected}) = 0.532 \pm 0.031$, this is in a network with average node degree = 18.7. Note that, as the k value increases, both the probability of a minimum node degree of k and the probability of k-connectivity decrease. For example, in Figure 4 (a), the $P(d_{min} \geq 2) = 0.522$ in a network with 150 nodes, while $P(d_{min} \geq 3) = 0.128$. The results in Figure 4 (a) and 4 (b) illustrate that the probabilities of achieving a minimum node degree of k and k-connectivity increase as the network density increases. The estimated boundaries of upper and lower limits are also plotted in Figure 4 as broken lines using estimations in [32] with the parameters adopted in our simulation. It shows that the bounds are not a good estimation of $P(k\text{-connected})$ in the tested network conditions. Therefore, we observe that the assumption of minimum node degree being k implying k-connectivity is not valid in sparse networks and even in the network with medium density.



(a) $k = 2$



(b) $k = 3$

Figure 4. Probabilities of minimum node degree and *k*-connected in different network densities

3.2.2 Average Node Degree and *k*-Connectivity

We now study the relationship between average node degree of the network and *k*-connectivity (i.e., number of disjoint paths). In next set of simulations, we used ns2 to generate random 50 node ad hoc network topologies. Again, we use a transmission range of 250m in area of 1000×1000 m^2. For values of k = 2, 3, 4, and 5, we generate random network topologies until 100 connected topologies are found with minimum node degree k. We analyze the 100 topologies found for each k value of 2, 3, 4, and 5. Table 1 shows the observed and calculated data from the obtained topologies. Specifically, Table 1 includes the average minimum node degree (*Ave Min ND*), the average minimum number of disjoint paths (*Ave Min DJP*), average node degree (*Ave ND*), and average number of disjoint paths (*Ave DJP*) for each of the 100

network topologies. *Ave Min ND* is computed by average minimum node degrees of 100 topologies. *Ave Min DJP* means the average of the minimum number of disjoint paths of 100 topologies. *Ave ND* is an average of the average node degree from each topology and *Ave DJP* is an average of the average number of disjoint paths from each topology.

According to Table 1, the average number of disjoint paths is always lower than the average node degree and they do not increase greatly as the k value increases. When the average minimum node degree changes from 2.33 to 5.00, the average minimum number of disjoint paths does not change significantly (it goes from 1.41 to 2.21). The difference between average minimum node degree and average minimum number of disjoint paths slightly increases when the k value is larger. The difference is 39.5% at $k = 2$ and 55.8% at $k = 5$. This indicates that at any given k value, on average, the minimum node degree requirements do not ensure k-connectivity.

Table 1. Minimum and average node degree and number of disjoint paths in different minimum node degree requirements

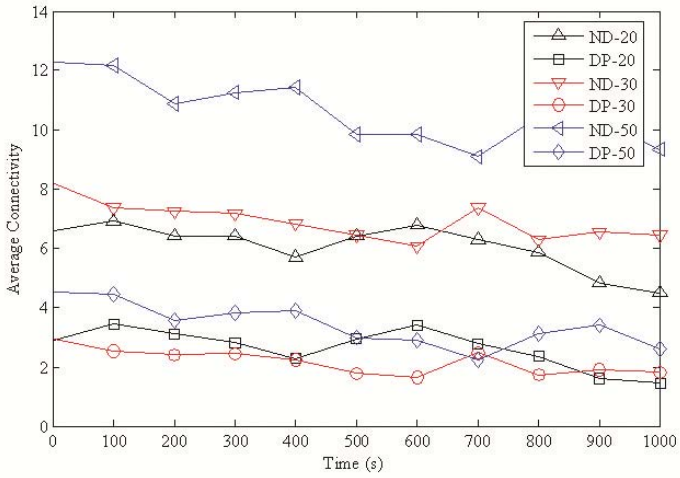
Req'd ND	Ave Min ND	Ave Min DJP	Ave ND	Ave DJP
2	2.33	1.41	7.65	3.89
3	3.09	1.50	7.63	3.86
4	4.02	1.83	7.83	4.03
5	5.00	2.21	8.39	4.34

Ave Min ND – Average of minimum node degree of 100 satisfied topologies; ***Ave Min DJP*** – Average of minimum number of disjoint paths of 100 satisfied topologies; ***Ave ND*** – Average of the average node degree of each topology; ***Ave DJP*** – Average of the average number of disjoint paths of each topology

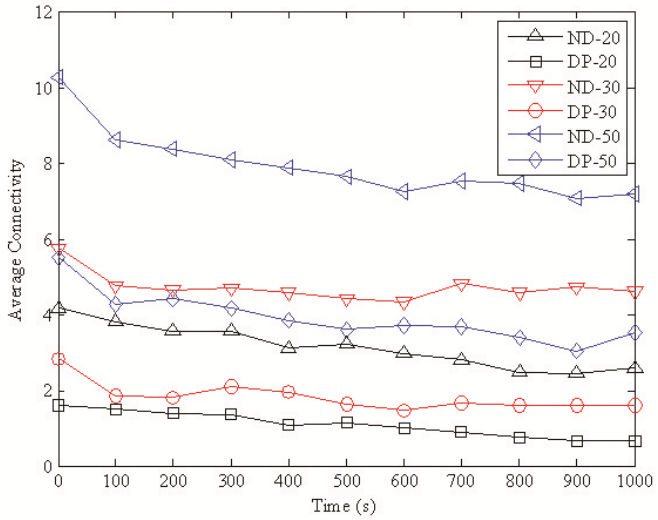
3.2.3 Average Node Degree and k-Connectivity under Mobility

In this simulation, we add the node mobility in order to observe the average node degree and the average number of disjoint paths in sparse ad hoc networks with mobility. We use the BonnMotion simulation tool with two mobility models – the Reference Point Group Mobility (RPGM) Model and Random Waypoint Mobility (RWM) Model. The maximum speed is 10 m/s and minimum speed is 0.5 m/s. A deployment area of 1000×1000 m^2 is chosen with 20, 30, and 50 nodes to understand the impact of node density. The transmission range is fixed at 250m and we use the unit disk model for link connectivity (i.e., two nodes have a link of they are within transmission range of each other and no link exists otherwise). The simulation is run for 1000 seconds and we capture a snap shot every 100 seconds. We observe the sequence of 10 topologies starting from the initial topology at time 0. Based on the node positions in each snapshot, we can obtain the network connectivity. First, we evaluate the relationship between the average node degree (ND) and average number of disjoint paths (DP) of all pairs of nodes with the two mobility models, RPGM and RWM, at different node densities (20, 30, and 50 nodes in the 1000×1000 m^2 area). We compare these two averages in Figure 5. We compute the average nodal degree and the average number of disjoint paths for each topology captured every 100 seconds in different scenarios. In RPGM model as shown in Figure 5 (a), the average number of disjoint path does not show significant difference among 3 different network densities, (i.e., 20, 30, and 50 nodes in 1000×1000 m^2 area) while the RWM model shows that 50 nodes in area of 1000×1000 m^2 holds averagely twice more number of disjoint paths than 20 or 30 nodes network has as shown in Figure 5 (b). This phenomenon may be caused by the node mobility model. In

RPGM, once the node involves in group, it moves along with its group leader and this prevent from creating connection between nodes that are in different group.



(a) RPGM Model



(b) RWM Model

Figure 5. Average node degree and average number of disjoint paths in different network density, 20, 30, 50 nodes in 1000x1000 m^2, over 1000 seconds of simulation time in two mobility models, (a) RPGM and (b) RWM; ND – Average node degree, DP – Average number of disjoint path

According to Figure 5, the average number of disjoint paths is always lower than the average node degree. The difference between the two averages increases when the network density increases. Another observation is that the behaviors of both average node degree and the average number of disjoint paths have similar tendencies over the simulation time. When the average node degree increases, the average number of disjoint path increases and it decreases when the average node degree decreases in time.

From these results, the average node degree always has a higher number than the average number of disjoint paths; the average number of disjoint paths is about 50% less than the average node degree. In other words, the average number of disjoint paths is smaller than the average node degree at all times. *The simulation results indicate that maintaining an average node degree cannot guarantee an equal average number of disjoint paths.* The average number of disjoint paths is approximated to only 50% of average node degree.

We examine the partition check every 100 seconds during simulation time of 1000 seconds in order to compute how many times the network partitioned during the simulation for both mobility models. RWM partitioned the network averagely 2.7 times with a maximum of 6. In case of RPGM, since nodes are gathered around the leader initially, network is partitioned already at time 0s. Then, we compute how many time the number of partitioned network during the simulation. Average is 8.4 with a maximum of 10. From this observation, it is found that the average node degree cannot provide network partition information. Even if the network is partitioned during the simulation, the average node degree does not change noticeably and neither does the average number of disjoint paths. If the nodes move too fast, the frequency of the network topology change is very high. Therefore, the randomness of the network is more considerable in the high mobility network.

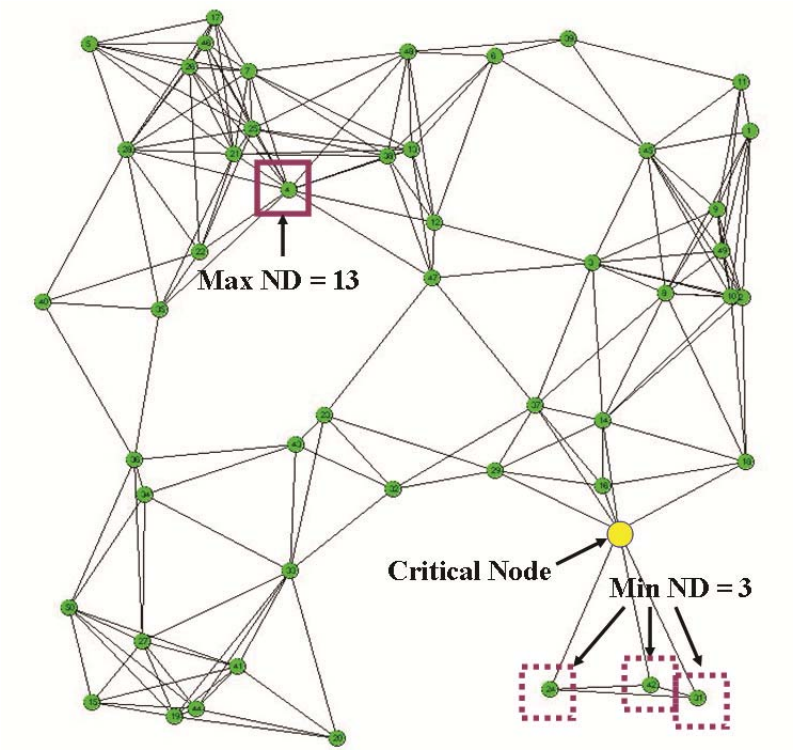


Figure 6. Sample Fifty Node Network Topology with Critical Node

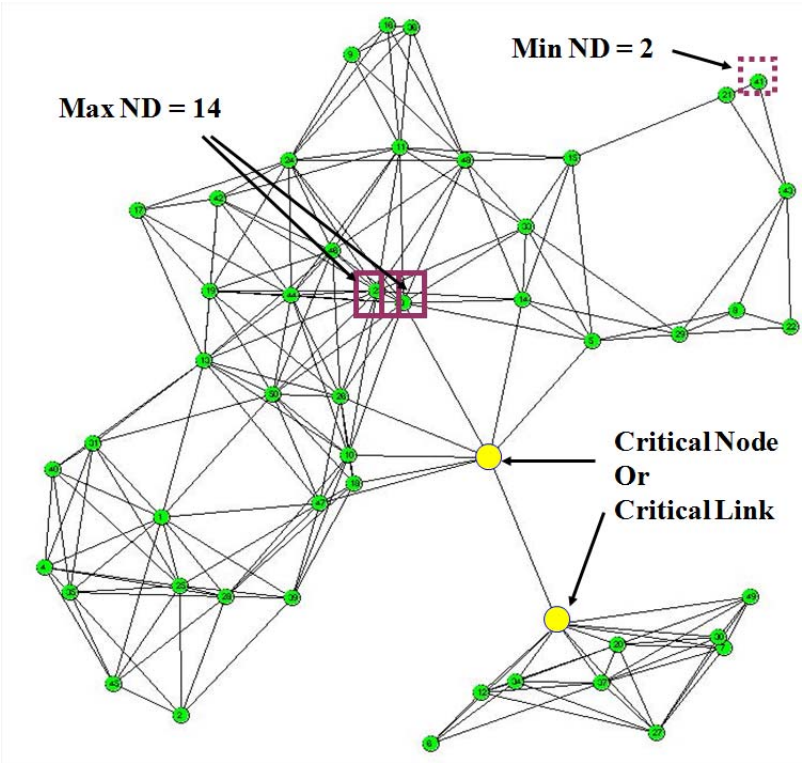


Figure 7. Sample Fifty Node Network Topology with Critical node and Critical link

3.2.4 Discussions

From above results, we find that neither the average nor the minimum node degree can represent the number of disjoint paths (*k*-connectivity). The problem is more severe than simple inability to predict *k* -connectivity as illustrated in Figures 6 and 7. In Figure 6, the minimum node degree is 3 and the maximum node degree is 13. However, there exists a node that partitions the network i.e., a critical node. In Figure 7, the maximum node degree is 14 and minimum node degree is 2. But there still exist critical nodes or critical links. In such cases, the network is partitioned when the link between critical nodes fails or the critical node itself fails. This shows that if the critical nodes are connected in one hop then, it is also bridge link or critical link. This lead us that all critical links are included if we find all critical nodes.

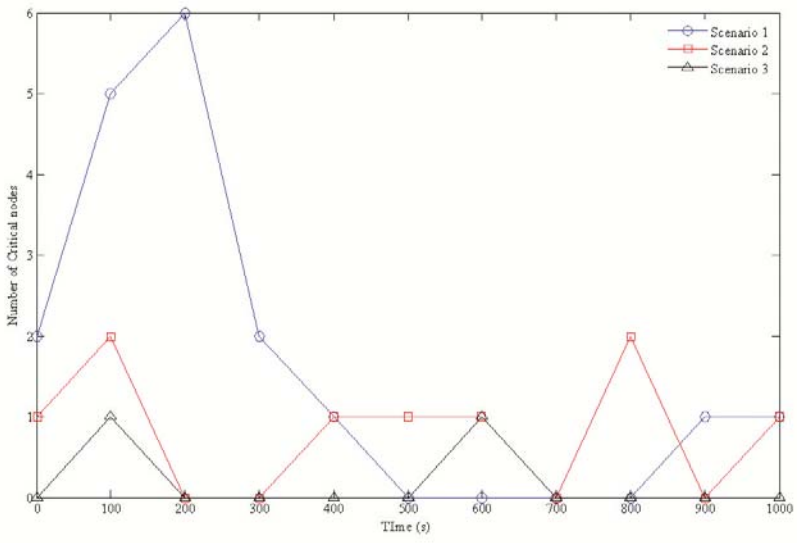


Figure 8. Number of Critical nodes behavior vs. simulation time

If we consider that the network is alive as long as the network is not partitioned. Based on this definition, the critical node is a weak point in the network and the network can be more

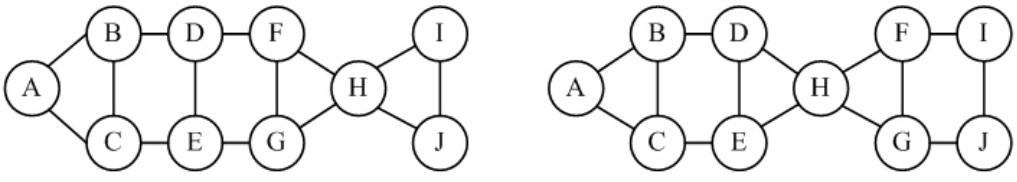
survivable when those critical nodes are strengthened in many ways such as increasing transmission range, recharge the battery, and etc. This definition also indicates that either average or minimum node degree does not suitable survivability metric because it cannot sense the network partition. We consider the network partition as a network failure in our new approach. Weak point of the network is considered and it could be either bridge link or articulation node. We provide 3 of 50 nodes network scenarios in 1000x100 m^2 with 250 m transmission range. Random Way Point Mobility model is chosen for this simulation due to the difficulty of being connected in Reference Point Group Mobility model. We collect 3 scenarios that the network is connected (no partition) during the simulation time of 1000 seconds. Figure 8 shows that the number of critical nodes during simulation time for all 3 scenarios. Figure 8 shows that the number of critical node is not consistent or pattern, but random. This result indicates that the network has to observe the critical nodes periodically for the survivability of the network at each time due to its topology changing. To identify the critical nodes, several algorithms are introduced [37-39] and they can be used to find critical nodes to make the network more survivable by strengthen them to prevent network partition. The faster nodes mobility is, the more frequent network topology changes. For the critical point study, therefore, we assume that the network with no mobility or slow enough mobility in the rest of this paper.

3.3 IMPACT OF CRITICAL NODE FAILURE

Here, we found out that the critical point plays a big role in network connectivity. In this section, we study the comparison of the impacts on network performance from failures of the maximum

degree nodes and critical nodes. The network performance measure we use for comparison study is packet loss rate.

When the critical node fails, the network is partitioned into more than two clusters. Among existing traffics, if the source and destination nodes are associated in different clusters after critical node failure, packets will be lost. Here, we introduce the partition rate that is the ratio of the number of nodes in partitioned small cluster to the number of total nodes of the network before partition. For example in Figure 9, two nodes are isolated when the node H fails in (a) and its partition rate is 20%. Similarly, the topology in Figure 9 (b) has partition rate of 40%. Then, partition rate may affect the packet loss. When the partition rate increases, the more possible traffic sessions exist between partitioned clusters. And this may increase the packet loss rate. For the network with n nodes, it has total of $n(n-1)$ possible traffic sessions. If the partition rate is r_p, the possible disconnecting traffics due to critical node H failure will be $2r_p n\left[\left(1-r_p\right)n-1\right]$. Therefore, the possible disconnecting traffics increases as the partition rate of r_p increases with fixed number of nodes n. For example in Figure 9, both topologies, (a) and (b), have 10 nodes and possible traffic is 90. But when critical node H fails, 28 traffic sessions in (a) and 40 traffic sessions in (b) are disconnected. Therefore, the larger partition rate may lead the higher packet loss rate with high probability.

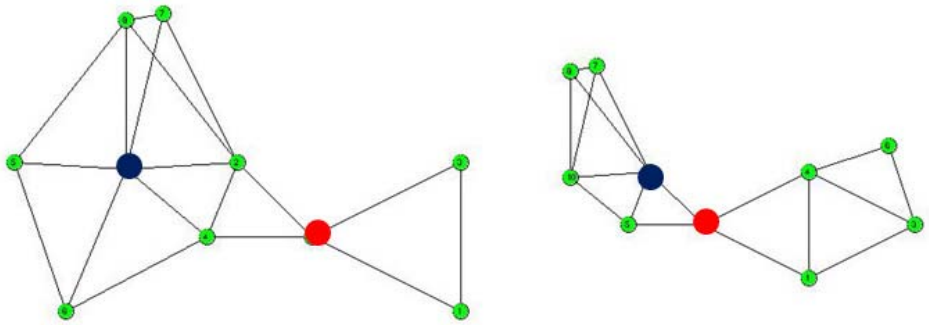


(a) 20% partition rate network (b) 40% partition rate network

Figure 9. 20% and 40% partition rates of the network when failure occurs

3.3.1 Simulation Study

In simulation study, we vary the partition rate with fixed network density. We select 20% and 40% partition rate and 10, 30, and 50 nodes for the network density over 5, 10, 20, 30, 40, and 50 maximum numbers of traffics. Traffics are 512 bytes of random Constant Bit Rate (CBR) with an interval of 0.25 seconds CBRs. NS2 is used to observe the packet loss. We fail the critical or maximum degree node at 400 simulation seconds out of total simulation time of 600 seconds. Thus, packet loss is observed during 200 seconds.



(a) 10 nodes density with 20% PR (b) 10 nodes density with 40% PR

Figure 10. 10 nodes network density topologies with 20% and 40% partition rates

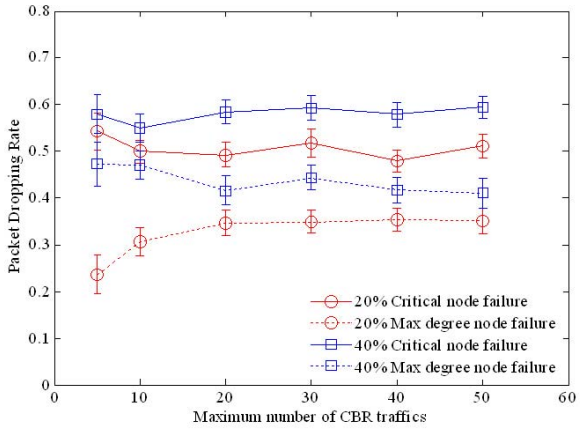
The topologies used for 10 nodes network density are shown in Figure 10. The topology in Figure 10 (a) indicates 20% of partition rate and (b) indicates 40% of partition rate. The red nodes represent critical nodes and blue nodes represent maximum degree nodes for both topologies. Figure 10 (a) isolates 2 nodes out of 10 nodes and Figure 10 (b) does 4 nodes out of 10 nodes when critical node fails. However, failure of maximum degree node in 10 nodes topology with 40% partition rate produces more bottle necks or critical points. This may increase higher packet loss rate when maximum degree node fails. All other topologies of 30 and 50

nodes do not produce bottle necks or critical points due to maximum degree node failure such as Figure 10 (a).

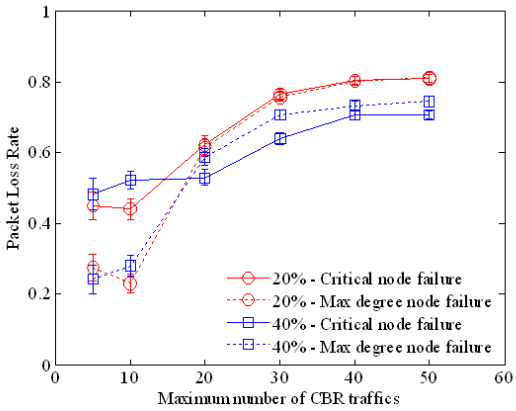
3.3.2 Results and Discussions

The simulation results are shown in Figure 11. The solid line indicates the packet loss rate when critical node fails and dotted line represents the packet loss rate when maximum degree node fails. Figure 11 (a) shows packet loss rate for 10 nodes network with 20% and 40% partition rate. The packet loss rate is higher when critical node fails for both cases. However, packet loss rate in 40% partition rate is higher than 20% partition rate in case of maximum degree node failure. As mentioned, 10 nodes topology with 40% partition rate creates more bottle necks or critical points when maximum degree node fails and it increase the packet loss when compare to that with 20%. Although maximum degree node failure produces more bottle necks or critical points, critical node failure is worse in packet loss. All other network densities also show that packet loss is more severe in critical node failure at low traffic load. The higher partition rate loses more packets and it becomes more significant when network is more dense at low traffic load by comparing Figure 11 (b) and (c).

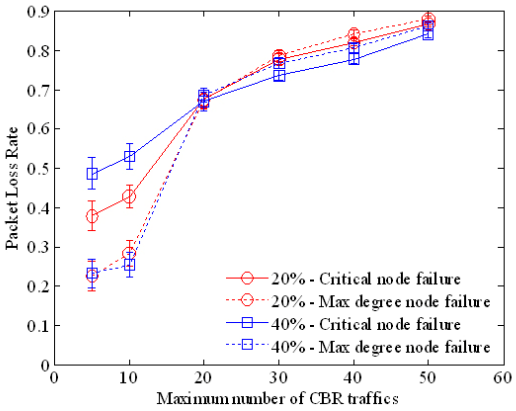
When the network density varies with fixed partition rate, it does not affect the packet loss rate except for 10 nodes network density because 5 maximum traffic loads is relatively heavy. This is shown in Figure 11 (d) and (e) and packet loss is similar for 30 and 50 nodes network in both 20% and 40% partition rate cases.



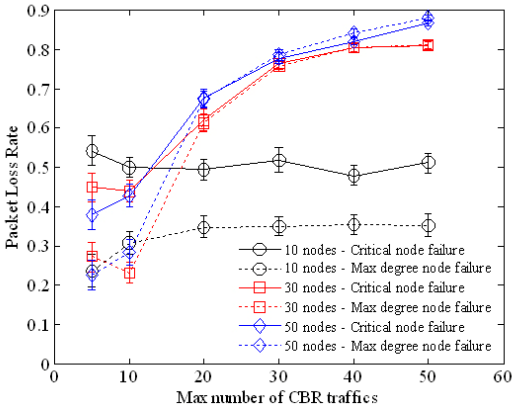
(a) 10 nodes network with 20% and 40% PR



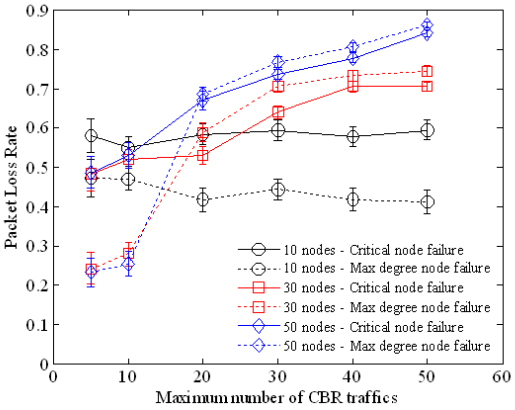
(b) 30 nodes network with 20% and 40% PR



(c) 50 nodes network with 20% and 40% PR



(d) 20% PR of 10, 30, and 50 nodes networks



(e) 40% PR of 10, 30, and 50 nodes networks

Figure 11. Packet Loss Rate (PLRs) in different partition rates, 20% and 40%, with same network densities, 10 nodes, 30 nodes, and 50 nodes

3.4 CONCLUSIONS

The connectivity is fundamental principle of the survivability in MANETs. Node degree represents the connectivity of the network in general. However, node degree information cannot guarantee the network to be connected because critical node exists even with high node degree. Maximum degree node and critical node are compared in packet loss rate as a performance measure. It shows that the packet loss rate increases more when the critical node fails. Therefore, the network can be more resilient by finding the critical nodes and strengthening them. There are several algorithms to identify the critical nodes.

4.0 CRITICAL POINTS IDENTIFICATION ALGORITHMS AND STUDY IN MOBILE ADHOC AND SENSOR NETWORKS

Given a connected sensor or mobile ad hoc network (MANET), the weak or critical points of the topology are those links and nodes whose failure results in partitioning of the network. In order to effectively deploy techniques to improve the resilience of sensor networks and MANETs, one must be able to identify all the weak points of the network topology. We explore the network metrics such as node degree and usage in identifying critical points. Then, we propose new algorithms based on results from algebraic graph theory, that can find the critical points in the network for single and multiple failure cases, and network connectivity between neighbor nodes. Utilizing the algorithm using algebraic graph theory we present numerical results that examine how the number of critical points varies with nodal density. We also explore the impact of local network topology information on critical point identification.

In this chapter, several metrics are examined to indentify the weak points of the network. Then, two heuristic algorithms are introducing and studying the indentified weak points of the network.

4.1 WEAK POINTS OF THE NETWORK

The weak points of the network are the point that causes network partition when they fail. Based on graph theory, these points are called bridge link or articulation node as shown in Figure 3 in Chapter 4. Bridge link is a link that makes network partition when it fails and so does articulation node. There are several possible metrics to identify the weak points of the network. In this section, several possible metrics are introduced and evaluated in identifying weak points of the network. The evaluating metrics are minimum and maximum node degree, maximum usage node in primary paths of all pairs of the nodes, and maximum usage node in all disjoint paths of every pair of nodes.

First, bridge link and articulation node are examined their significance as a weak point in the aspect of network connectivity in this section. Then, possible metrics are testing to indentify the weak points of the network.

4.1.1 Significance of Bridge Link and Articulation Node

The weak points of the network are one of bridge link or articulation node. Node failure may occur due to node mobility, power depletion, jamming attack, and etc. while link failure may occur due to obstacles or excessive interference in homogeneous wireless ad hoc network. Here, the significance comparison between bridge link and articulation node in connectivity aspect is performed using simulation study in ad hoc network.

In simulation study, uniformly distributed 100 connected network topologies are randomly generated for each network density of 50, 65, 75, 85, 100, and 125 nodes in 1500 x 1500 m^2 where the connected network means that every pair of nodes has at least one path. The

connectivity is established using transmission range of 250m. Any pair of nodes in this range establish the direct link each other. Graph G is created based on network connectivity and the bridge links and articulation nodes can be identified using algorithm in [39]. The number of each bridge links and articulation nodes is counted and we compute the articulation ratio to the total number of identifying critical points, where the number of total considered nodes, N_C, includes the nodes in bridge links, N_B, and the articulation nodes, N_A (i.e., $N_C = N_B + N_A$). We count 2 as a number of nodes in bridge link, N_B, for each bridge link because each bridge link contains two end nodes. The articulation ratio to the total number of identifying critical points is shown in Equation (6).

$$Ratio_{articulation} = \frac{N_A}{N_C} \tag{6}$$

Based on simulation study, each ratio is calculated for 100 topologies and they are averaged with 95% confidence in each network density as observed in Figure 12. The average articulation node ratio increases as the network density increases. At 75 nodes density, the ratio of articulation nodes to total number of critical nodes is getting larger than 50% and it increases up to around 70% at 125 nodes density. These results indicate that an articulation node is more common weak points at high network density.

Another observation from this simulation is the position of the weak points, bridge links and articulation nodes, at dense network. Most of bridge links and articulation nodes are located on the edge of the network in dense network, such as 100 and 125 nodes. Generally, the more number of nodes in same network area creates more abundant connectivity. As a result of this abundant connectivity, the core part of the network is obviously well connected and it may avoid creating critical points with a high probability. Thus, the most of critical points in dense network

are positioned in border area of the network and they are more likely articulation nodes rather than bridge links. However, this will be studied more detail in later part of this chapter.

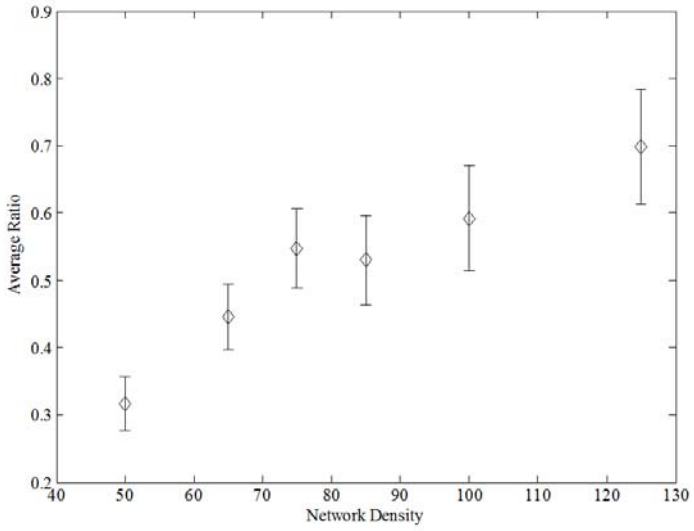


Figure 12. Average Ratio of the number of articulation nodes to the total number of critical nodes with 95% confidential in different network densities

4.1.2 Identifying Critical Node Methods

In wired networks, the concept of identifying critical network nodes has recently been investigated from an infrastructure protection standpoint [42]. This literature focuses on which nodes will have the most impact on the network (in terms of traffic loss) and a variety of heuristics have been proposed to identify the critical nodes such as, maximum degree nodes, and maximum traffic nodes. In this section, some simple heuristics in identifying critical nodes are examined for their effectiveness, specifically we consider, the maximum node degree (Max ND), minimum node degree (Min ND), most heavily utilized nodes (largest number of shortest path routes), and nodes having the most backup path routes passing through them. The backup path routes are node disjoint with the shortest path route for each pair of nodes.

In heuristics evaluation, we test 100 connected random topologies, which contain at least one critical node, where the area of 1000x1000 m^2 with 50 nodes and 250m of communication range. Then, we compute the ratio of the number of detected critical nodes to the total number of critical nodes in Equation (7).

$$Ratio = \frac{Number\ of\ Detected\ Critical\ Nodes}{Total\ Number\ of\ Critical\ Nodes} \times 100(\%) \quad (7)$$

Table 2. Percentage of correct critical node detection

	Max ND	**Min ND**	**Heavy Usage**	**Greatest Backup**
Correct Alarm	12.33%	0.83%	31.95%	10.50%
95% Confidence interval	±5.88%	±1.17%	±8.00%	±4.93%
False Alarm	87.67%	99.17%	68.05%	89.50%

Max ND – 3 highest node degree nodes; ***Min ND*** – 3 smallest node degree nodes; ***Heavy Usage*** – 3 heavily utilized nodes in primary route; ***Greatest Backup*** – 3 highly used nodes in backup route

Table 2 shows the average percentage of correct detection in the 100 topologies using the metrics of maximum and minimum degree and the usage in primary and disjoint routes with 95% confidence intervals. In this observation, best three nodes in each metrics are chosen for evaluation. It shows that the metric of heavily utilized in shortest routes can identify the critical nodes better among them. However, its detection rate is still very low, 32 % and it infers that none of them is a good critical node identifying method.

4.2 HEURISTIC ALGORITHMS

Two heuristic algorithms to identify the critical points in the network are proposed in this section. The first technique is based on adapting a graph theoretic test for network connectivity to the critical point identification problem. The second approach is based on utilizing the connectivity between neighbor nodes.

4.2.1 Algorithm I

Graph theory is implemented in Algorithm I to indentify the critical node in the network. In graph theory, algebraic connectivity is used to test each node for its criticalness. Algebraic connectivity is the second smallest value of the Laplacian matrix of network connectivity, so called adjacent matrix. Consider an arbitrary MANET or sensor network topology of N nodes. The network topology can be represented by an N x N adjacency matrix. Let $A(t)$ denote the adjacency matrix at time t as shown in equation (8).

$$A(t) = \begin{bmatrix} a_{11}(t) & a_{12}(t) & \cdots & a_{1N}(t) \\ a_{21}(t) & a_{22}(t) & \cdots & a_{2N}(t) \\ \vdots & \vdots & & \vdots \\ a_{N1}(t) & a_{N1}(t) & \cdots & a_{NN}(t) \end{bmatrix} \tag{8}$$

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.

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., $a_{ij}(t) = 1 \rightarrow a_{ji}(t) = 1$). Let $d_i(t)$ denote the degree of node $i \in N$ at time t (i.e., $d_i(t)$ equals the number of links to other nodes from node 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., $D(t) = diag\big(d_i(t)\big)$).

$$D(t) = \begin{bmatrix} d_1(t) & 0 & \cdots & 0 \\ 0 & d_2(t) & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & d_N(t) \end{bmatrix} \tag{9}$$

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

$$L(t) = D(t) - A(t) \tag{10}$$

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 [43], 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 [43] is equal to the number of connected components of the network. 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 link or nodes) in the network. Let T denote a set of points (i.e., links, nodes or combination 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.

Table 3. Pseudo code of Algorithm I

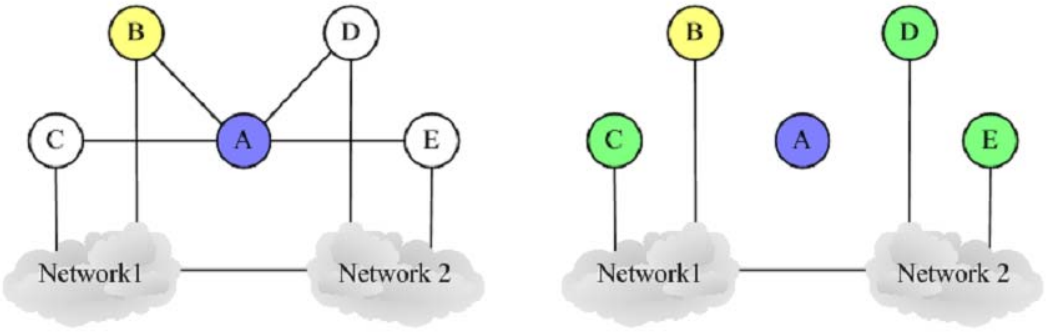
Steps of Algorithm I	
Step 1	Test point $i \in T$ is chosen to check its critical status
Step 2	Eliminate test point i from the adjacency matrix A and recompute the nodal degrees in D. Specifically if i is a node then remove row i and column i from A and adjust D, if i is a link then set the appropriate 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 then i is a critical point, otherwise i is not critical and the network is still connected
Step 5	Choose 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 [50]). Also, note that many efficient

algorithms exist for eigenvalue computation so the computational burden on nodes is not excessive.

4.2.2 Algorithm II

Algorithm II utilizes the routing layer protocol to test the connectivity of the network. It performs locally at a node as a self-test and utilizes the network routing algorithm such as Dijkstra. The basic idea is to test a link or node for criticality by deleting all routes through the test point and seeing if alternate routes exist. In this algorithm, it assumes all nodes update their neighbors' information periodically. Figure 13 shows a simple example of this algorithm.



(a) Node A performs Algorithm II (b) Disconnect A from all its neighbor nodes

Figure 13. Algorithm II: Self-critical testing at node A

In Figure 13, node A has 3 neighbor nodes, B, C, and D. If node A executes Algorithm II critical test, it chooses one of its neighbor nodes, for example B in Figure 13. Node B computes the shortest path, not passing through node A, to all other neighbor nodes of A, (i.e., C, D, and E). If there are paths from B to all other neighbor nodes shown in Figure 13 (b) via network 1 and 2, A is not a critical node. If B cannot reach any one of A's neighbor nodes (C, D, or E) such that network 1 and 2 are not connected, then A is a critical node. In order to find the path not passing

through node A, node A in effect disconnects itself to its neighbor nodes as shown in Figure 13 (b). The pseudo code of Algorithm II is shown below in Table 4.

Table 4. Pseudo code of Algorithm II

	Steps of Algorithm II
Step 1	Test point $i \in T$ is chosen to check its critical status
Step 2	Logically disconnect *i* from the network routing information. Specifically if *i* is a node then remove all links to and from *i,* if *i* is a link then remove the link
Step 3	Check if there exists a route around the test point *i*. Specifically if *i* is a node then choose one of *i*'s neighbor nodes (e.g., B) and run the network routing algorithm to determine if a route exists from the chosen node (e.g., B) to all other neighbor nodes of *i*. If testpoint *i* is a link connecting two node j and k, then run the network routing algorithm to see if a route exists connecting j to k
Step 4	If alternate routes exist then testpoint *i* is not critical, otherwise *i* is critical
Step 5	Choose next test point $i \in T$ and go back to step 2

The accuracy of Algorithm II in determining critical points will depend on the routing algorithm used and the amount of topological information available. If global routing information is available then Algorithm II can find all critical points. Otherwise, the correctness of identifying critical points using Algorithm II depends on the network traffic condition.

4.2.3 Limitation and Comparison of Heuristic Algorithms

The time complexity of algorithm I 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 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.

The time complexity of Algorithm II is determined by shortest path finding algorithm. For example, if the Dijkstra algorithm is used for shortest path finding, its time complexity is $O(n^2)$ on a graph with n nodes and m edges. Although algorithms have a same time complexity, we use Algorithm I for further studies since Algorithm I has more ability to provide more information about the network by graph theory such as eigenvalues and eigenvectors of Laplacian matrix.

4.3 CRITICAL NODE STUDY

We implement heuristic algorithm I in MATLAB to identify the critical nodes in order to study them. In this section, we study the behavior of critical nodes comparing to average node degree and average number of disjoint paths first. Then, we examine the number of critical nodes in different network density including multiple critical nodes, where the multiple critical nodes is the combination of nodes that the network partitions when they fail at the same time.

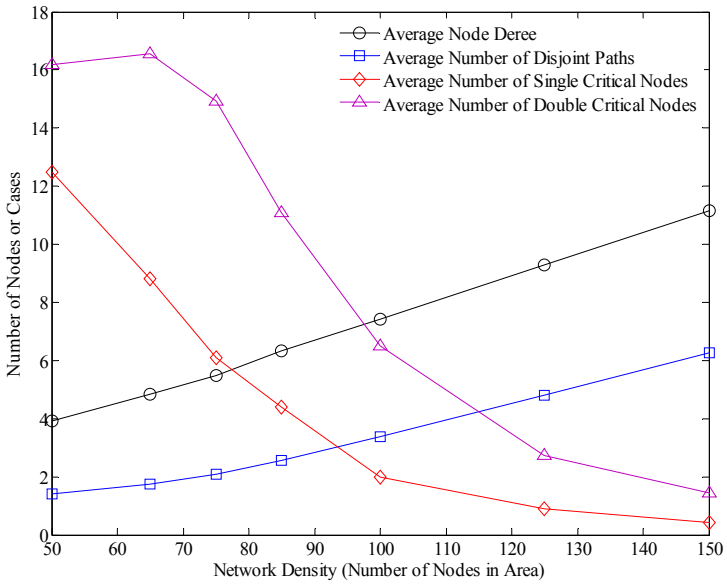


Figure 14. Average Number of Single Critical nodes, Average Number of Double Critical Nodes, Average Node Degree and Average Number of Disjoint Paths versus the Network Density

4.3.1 Number of Critical Nodes Behavior

In this study, Algorithm I is used to detect the critical nodes, which is implemented in MATLAB. The behavior of the number of critical nodes is examined in different network densities. Network topology is randomly generated with different number of nodes (i.e., 50, 65, 75, 85, 100, 125, and 150) in a $1500\times1500m^2$ network area. The nodes are randomly and independently distributed over the network area with the (x, y) coordinates determined according to two independent uniform [0-1500] random variables. It is assume that all nodes are identical with 250m transmission range. For each node density, we randomly generate topologies until 100 connected topologies are generated. 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. The double critical nodes can be detected by replacing testing points with all combinations of two nodes in Algorithm I.

These metrics are then averaged over the 100 topologies for each network density and plotted in Figure 14. 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.

4.3.2 Positions of Critical Nodes

The critical node is a node that partitions the network caused by its failure due to any reason. Thus, the critical node is considered as a connectivity wise vulnerable point of the network. It is questioned how the critical node locates over the network and how they varies in network density. We perform a simulation study to answer this question. The network topologies are randomly and independently generated and at least 1-connected. The number of nodes (50-, 75, 100, 125, 150, 175, 200) are uniformly distributed over the area of $1500\times1500m^2$. 250m of the transmission range for the network connectivity. In each network topology, the critical nodes are identified. First, the number of critical nodes is counted over 1000 topologies in each network densities. Table 5 illustrated these counted number of critical nodes for each network density. The number of critical nodes greatly decreases.

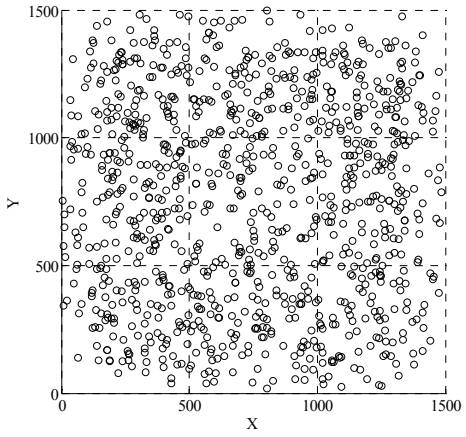
In simulation, topology is randomly generated with uniformly distributed nodes in 50, 75, 100, 125, 150, 175, and 200 nodes in area of 1500x1500 m^2. 250 m of transmission range is used

for the direct connectivity. In 50 to 150 nodes network, 1000 random topologies are generated to observe the behavior of critical nodes. 3000 random network topologies are generated for 175 nodes network and 5000 for 200 nodes network in order to obtain enough number of critical nodes for trustworthy observation. Table 5 shows the number of critical nodes obtained from each node density.

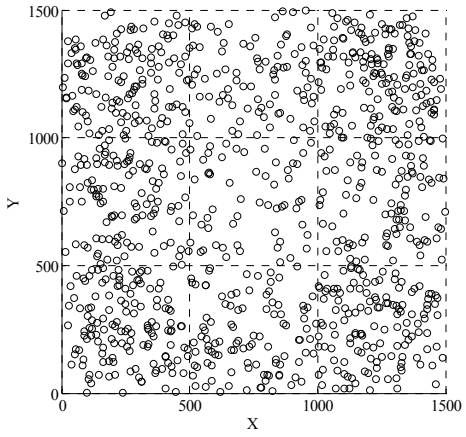
Table 5. Number of obtained critical nodes and topology generation for each density

Nodes network	Obtained critical nodes	Observed topologies
50	12,648	1000
75	6,094	1000
100	2,263	1000
125	945	1000
150	405	1000

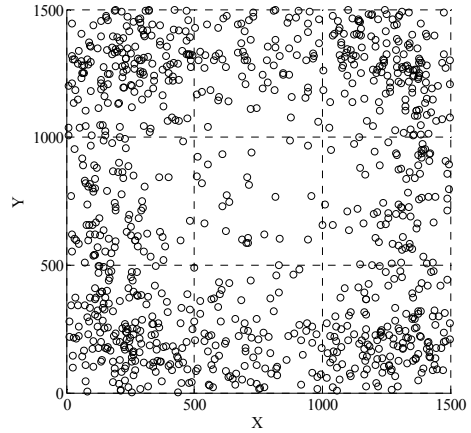
Next, the locations of the critical nodes are examined. In this study, we plot the critical node identified from above random topologies.



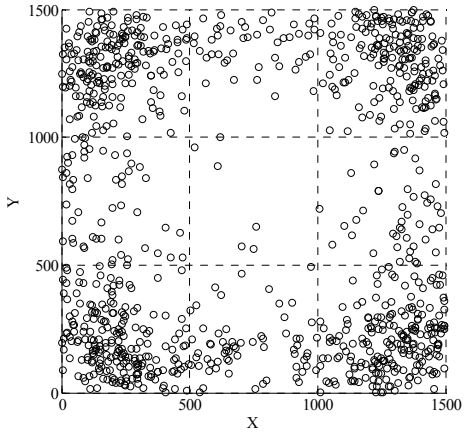
(a) CR_N locations of 50 nodes network



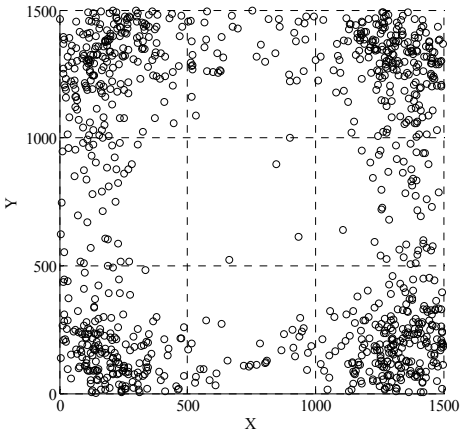
(b) CR_N locations of 75 nodes network



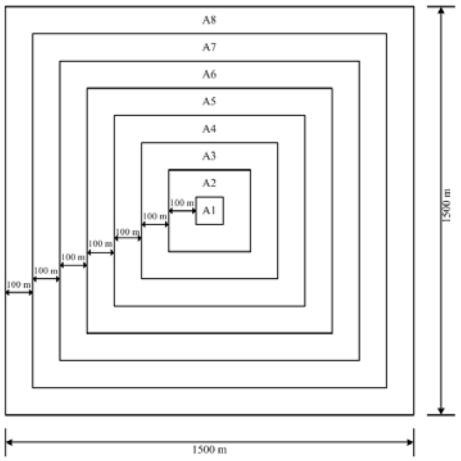
(c) CR_N locations of 100 nodes network



(d) CR_N locations of 125 nodes network



(e) CR_N locations of 150 nodes network



(f) Dividing sections

Figure 15. Critical Nodes Locations of 50, 75, 100, 125, and 150 nodes networks over the area of 1500×1500m^2 and 8 Sub-areas

In Figure 15, (a) through (e) represents the critical nodes for each network density. These figures indicate that the critical nodes are locating more edge area of the topology as the network is denser. For more detail observation on the position of the critical nodes in network area, we divide the area into 8 sections as shown in Figure 15(f). At every 100 m from the center point, it forms the rectangle and it divides the 1500×1500 m^2 area into 8 sub-areas, from A1 to A8. The

number of critical nodes is counted for each sub-area is calculated by dividing the counted number of critical nodes by total number of critical nodes.

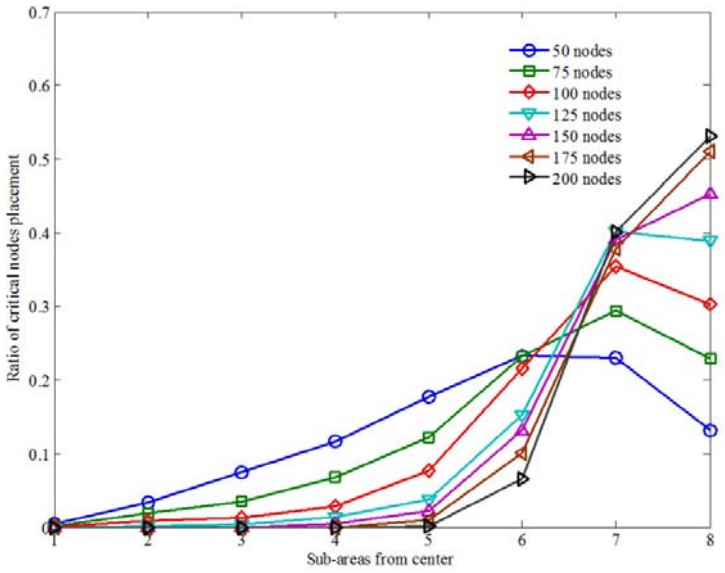
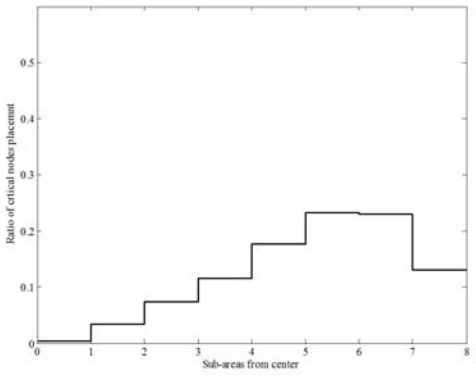
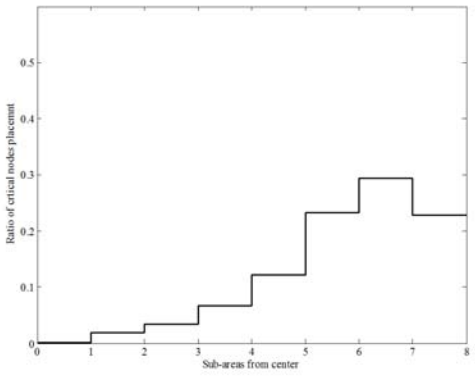


Figure 16. Ratio of critical nodes in sub-area divided starting from center for each density as in Figure 15

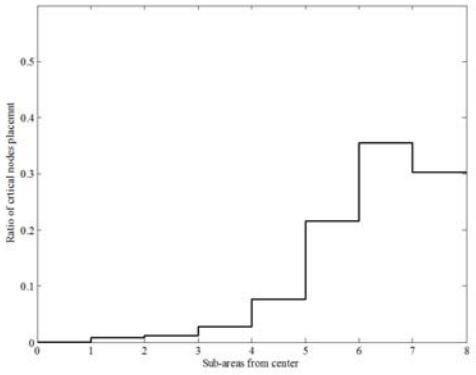
Figure 16 shows the ratio of critical node at each sub-area. Sub-areas are assigned as shown in Figure 15. Higher number of sub-area means that the distance is further from the center of the area. The result indicates that the possible critical node placement tends to move outward from the center as the network density increases. For 200 nodes network, most of critical nodes are located close to border area. At 50 nodes network, critical node positioning possibility decreases when it moves into the center of the network area. Next Figure 17 shows the ratio for each network density. Critical node tends to place farther from the center and this tendency becomes more ensured. The probability is definitely higher at the border area.



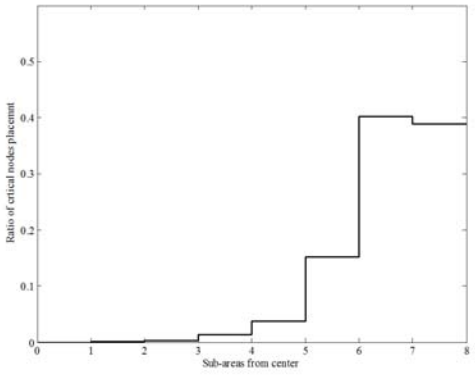
(a) 50 nodes network



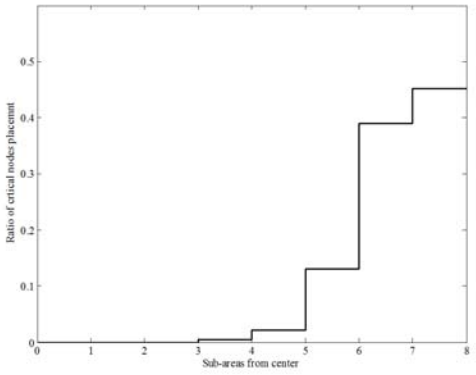
(b) 75 nodes network



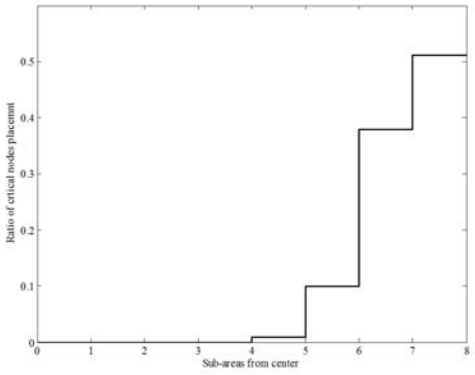
(c) 100 nodes network



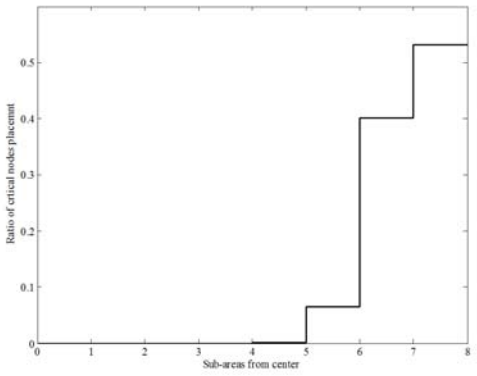
(d) 125 nodes network



(e) 150 nodes network



(f) 175 nodes network



(g) 200 nodes network

Figure 17. Probability of critical node placement at divided sub-areas for each network density

Next, we observe how the critical node failure affects on the portion of the partitioned network. The ratio of portion of the partitioned minor network is calculated by dividing the number of nodes in smaller size of partitioned network by the total number of nodes. This shows how risky the critical node is upon network partition. Figure 18 shows the average proportion of partitioned network with 95% confidence interval. At each network density, examined number of topologies is same as in Table 5. According to Figure 18, averagely 14% of the network is partitioned at 50 nodes network and it decreases steeply as the network size increases. The maximum is 50% at 50 nodes network with minimum of 2%. This indicates that when the network size is low, critical node is more important because it may cause major network partition. At 120 nodes network, the slope stats gentled and it begins converging. In 200 nodes network, the average partitioned network portion is less than 1%. This behavior can be explained that the critical node locates around the edge of the network when the network is dense enough.

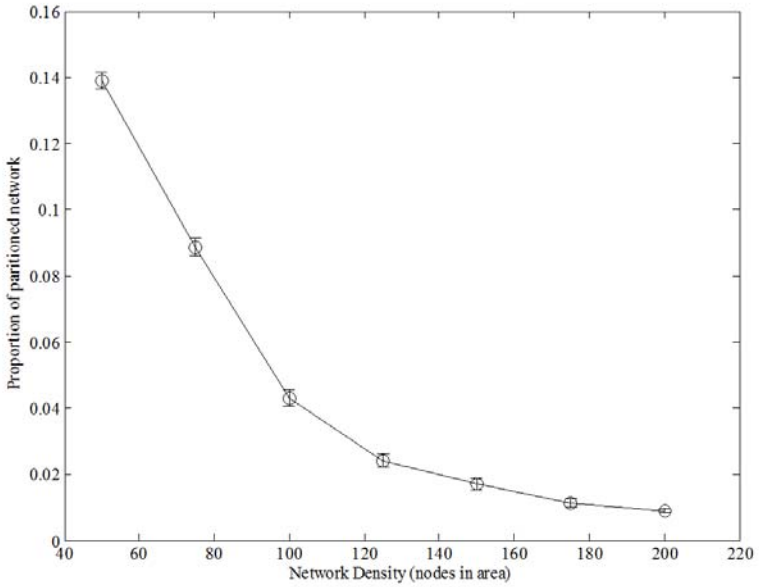


Figure 18. Portion of partitioned network due to critical node failure in each network density

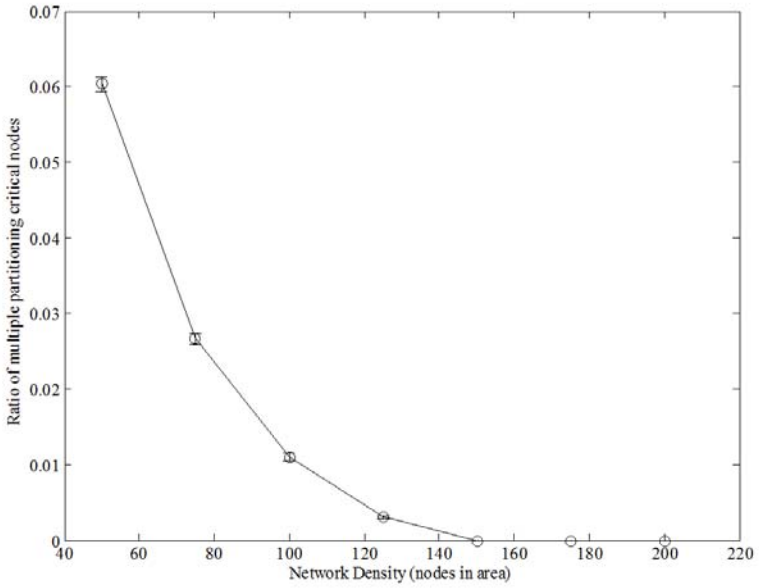


Figure 19. Number of occurrence of multiple clusters partition due to critical node failure in each network density

Next observation is about the number of partitioned clusters due to critical node failure. In this simulation, we measures how often the network partitioned into multiple clusters when a critical node fails. Simply, we count the number of multiple clusters partition occurrence when

the critical node fails. Figure 19 shows the ration of multiple clusters partitioning with 95% confidence interval. It points out that the multiple cluster partitioning is very rare. In 50 nodes network, only 6% of critical nodes partition the network into multiple clusters and the ratio converges to 0 at 150nodes network.

In this simulation study, we assume that only one critical node failure. We observe that the critical node is more likely locating over the border of the network and it is ensured when the network density increases. And the critical node is getting meaningless at dense network if certain portion of partition is allowed. In case of multiple clusters partitioning, a critical node failure does not play important role either. And the possible future works can be Consider multiple node failures (multiple critical nodes, 2, and 3) or Compare performance (packet dropping) with other metrics (max node degree, great usage node) in predetermined network topology.

4.4 CRITICAL LINK FINDING ALGORITHM

The proposed heuristic algorithms identify the critical nodes only. In this section, we introduce how to find the critical links utilizing known information by proposed heuristic algorithm I (i.e., critical nodes, eigenvectors, and node degree).

4.4.1 Critical Link

Critical link, also called as bridge link, is a link of whose failure partitions the network. In this session, we introduce how the critical link can be identified based on information obtained from

critical node identification such as critical nodes, eigenvectors, and node degree information. The advantage of this method is that it does not require any other additional steps or information in critical link identification process. The basic idea is described in theorem 1.

Theorem 1. *Both end nodes of critical links are also critical nodes if and only if the node degrees of both end nodes are greater than or equal to 2.*

$$d_i, d_j \geq 2 \ \wedge \ l_{ij} \in S_{cl} \ \rightarrow \ i, j \ \in S_{cr} \tag{11}$$

where l_{ij} is link between node *i* and *j* and S_{cl} and S_{cr} are Set of critical links and nodes respectively.

Proof. Assume a bridge link l_{ij}, of whose end nodes of *i* and *j*, connects the cluster A and B. The node degree of node *i* and *j* are obviously greater than or equal to 2 because each node are the member of each cluster and connected each other. In case of node degree of 2 for both end nodes, when node *i* or *j* fails, the link l_{ij} also fails. As a result of link l_{ij} failure, cluster A and B are disconnected. Therefore, node *i* and *j* are critical nodes.

However, theorem 1 is not *sufficient and necessary condition* which does not mean that all links, of whose end nodes are critical nodes, are not critical links (i.e., $i, j \in S_{cr} \wedge a_{ij} = 1 \nrightarrow l_{ij} \notin S_{cl}$) when node degree of end nodes are equal to 2. For larger node degrees than 2, there exists exceptional case. If third isolated cluster which connects both end nodes exists, the direct link between critical nodes is not a critical link. In other words, if there is multi-hop path between directly linked critical nodes, then the link is not critical. For example, Figure 20 illustrates the non-critical link between critical nodes case. In this network, node A and B are critical nodes and direct link is in between them and satisfy the condition of theorem 1. However, this is not critical link because the multi-hop path between them via node C and D exists. In

other words, critical node A and B share some part of network, which connect them without the direct link (i.e., l_{AB}).

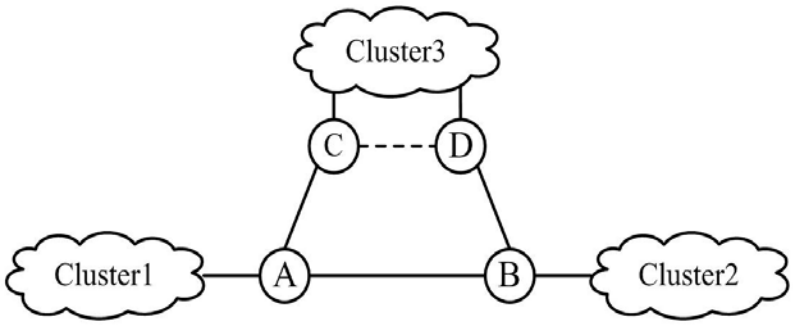


Figure 20. Example of non-critical link which has critical end nodes

In order to distinguish this case from the set of links of whose end nodes are critical nodes, theorem 2 is introduced.

Theorem 2. *If set of intersection between sets of isolated cluster which includes the other end nodes when one end node is removed is not empty, the link is not critical when node degrees of both end nodes are greater than 2.*

Proof. The proof is obvious that the direct link is not critical if there is multi-hop path between end nodes except direct link. Then, the intersection set of node sets of cluster which includes the other end node when one of end node is removed should not be empty.

Theorem 2 checks for 2 directly connected critical nodes share any common network by removing each other and checking for common nodes. For example, when node A is removed, the cluster containing node B has node C, D, and nodes in cluster 3 in Figure 20. Similarly, node C, D, and nodes in cluster 3 are also included in the cluster containing node A when node B is remove. Those node C, D, and nodes in cluster 3 are commonly shared by node A and B. Therefore, link AB (i.e., l_{AB}) is not critical link by theorem 2.

4.4.2 Critical Link Detection

Based on above theorems, we will introduce how critical links can be identified. Two types of critical links are identified, single critical link and double critical link. Single critical link is a link that makes network partition due to its failure where double critical link is a set of two links that makes network partition due to both link failures at the same time.

4.4.2.1 Single critical link detection

Using theorem 2, it can identify critical link when node degrees of both end nodes are greater than 2. In critical node identification algorithm, each testing node is removed and it computes and checks the eigenvalues. Then the eigenvectors are the additional information. According to graph theory, its well known property is that each element in eigenvector has its direction of corresponding node. This property provides the component nodes of each generated clusters by removing testing node. For example in Figure 20, when node A is testing for its criticality, node A is removed and the node set of cluster which node B is belongs to, $S_{CL\ni B}$ is {B, C, D, nodes in cluster2, nodes in cluster3}. Similarly, $S_{CL\ni A}$ is {A, C, D, nodes in cluster1, nodes in cluster 3}. Then, $S_{CL\ni B} \cap S_{CL\ni A} \neq \{\cdot\}$ and it is {C, D and nodes in cluster3}. This indicates that l_{AB} is not critical link. The last case of critical link is the last branch of the graph. If the one of end node has node degree of 1, then the link is critical. Therefore, theorem 1, 2, and last case of critical link can identify the critical links with no additional information besides the information obtained from critical node identifying by proposed algorithm.

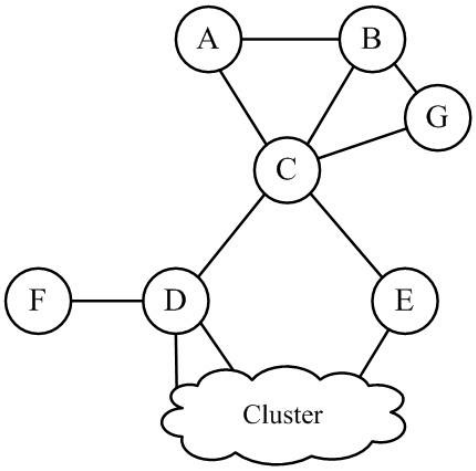


Figure 21. Special cases in selective testing node set

4.4.2.2 Double critical link detection

Identifying double critical links is high time consuming because large amount of all combinations of two links need to be checked. In order to reduce the computation time, we introduce a selective testing method. First, we narrow down the number of nodes in testing set and test for every combination of two links existing between nodes in the selective set. There are three node selection in double critical link identification; (1) single and double critical nodes and special cases such as (2) the nodes with node degree of 2 (i.e., $d_i = 2$) and its neighbor nodes and (3) all neighbor nodes of critical node if it has any critical node(s) among its neighbor. Links established by single or double critical nodes is the main link set that generates the single or double critical links. Case (2) is shown in Figure 21. The node degree of node A is 2 (i.e., $d_A = 2$) and link l_{AC} and l_{AB} are double critical links in this case. To identify this double critical links, node B should be added in testing set as well as node C which is already added as a critical node. The last special case (3) is shown in Figure 2, which is node E. Node C and D are added in testing set as a critical node. However, l_{CD} and l_{CE} are double critical links which is not able to be identified because node E is not in testing set. Node D and E is a double critical nodes but it is

not counted in double critical nodes since node D is already counted as a single critical node. This special case can be discovered by case (3).

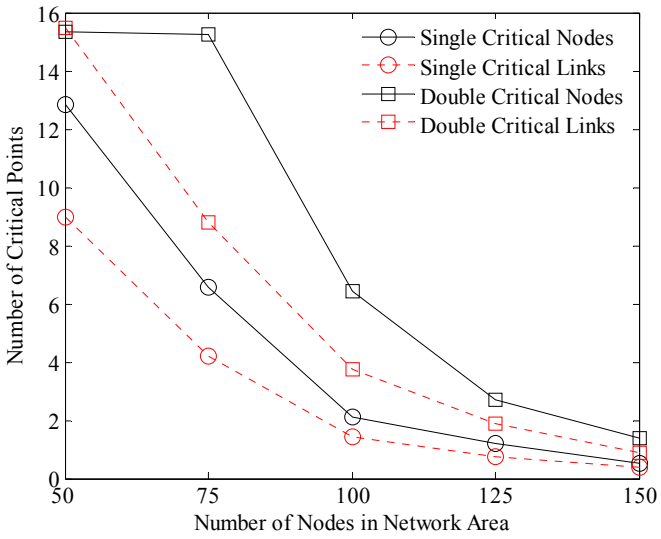


Figure 22. Comparison of single and double critical nodes and links

4.4.3 Numerical Study

We randomly generate 100 of 1-connected network topologies in different network densities such as 50, 75, 100, 125, and 150 nodes in the network area of 1500 by 1500 m^2, to test our critical point detection algorithm. Our algorithm identifies all critical points including single and double critical nodes and links. In here, we exclude the combination of 2 nodes or links including any single critical node or link from double critical nodes or links. We perform our algorithm on 100 randomly generated 1-connected network topologies over different network densities and count the number of critical nodes and links obtained by our algorithm. Then, we average the number of each critical point, node or link, and compare them in Figure 22. In sparse network such as 50 nodes, it shows that the double critical link is about to same risk as the single critical

node. Then, the critical node is more dominant as the network is denser. Double critical points, nodes and links, are more dominant than single critical points. Also, all critical points greatly decrease as the network is denser. For example, in 150 nodes network, average numbers of single critical node and link are 0.5 and 0.37 respectively whereas those of double critical nodes and links are 1.39 and 0.87 respectively.

Table 6. Computation Time Comparison

Network Density	**Non-Optimized Method**		**Optimized Method**	
	Mean Time (s)	CI (95%)	Mean Time (s)	CI (95%)
50	4.6769	1.2050	3.3536	1.2863
75	54.2809	9.3008	8.0044	4.6831
100	314.5273	39.3822	14.2225	2.5760
125	∞	N/A	34.1985	2.1797
150	∞	N/A	72.8496	3.1627

The computation time to identify the single and double critical links is greatly reduced by our modified algorithm, which uses the obtained information from critical node finding with reduced testing node set. The computation time of non-optimized and optimized are averaged out of 100 topologies in 50, 75 and 100 nodes networks. The total computation times identifying single and double critical links are measured and averaged with 95% confidence interval for non-optimized and optimized findings. These mean and 95% confidence interval of computation times are compared in Table 6. Non-optimized finding method takes very long time in denser network (i.e., 125 and 150 nodes network). In sparser network, the computation time of non-

optimized takes longer than optimized one. The mean computation time difference increases when the network is denser. For example, difference of mean computation time between non-optimized and optimized methods is 4.6769s – 3.3536s = 1.3233s in 50 nodes network. Similarly, it is 46.2765s in 75 nodes network and 300.3048s in 100 nodes network.

4.5 MULTIPLE CRITICAL POINTS

Multiple critical points is the combination of points whose simultaneous failures makes the network partitioned. In this study, we consider the combinations of nodes for the multiple critical points.

4.5.1 Multiple Critical Nodes

The multiple critical nodes is simply the combination of the nodes whose simultaneous failures partitions the network. If the single critical nodes exist, the network is 1-connected. For example, the network is partitioned if the single critical node fails. Besides of single critical nodes failure, the network still has a chance for the partitions when the certain combinations of multiple nodes fail at the same time. In Figure 23, for example, node H is a single critical weak point in the network and the single critical node is not included in multiple critical nodes. When the double critical nodes are considered, double critical nodes combinations are nodes AC, DG, and GH. In case of triple, any combination including nodes AD, DG, and GH are not considered. The, triple critical nodes combinations are nodes ADE, AEF, CEG, and DEG. Therefore, the number of single, double, and triple critical nodes is 1, 3, and 4, respectively.

Algorithm I is utilized to identify the single and the combination of multiple critical nodes. In double critical nodes testing, 2 different nodes are selected and Algorithm I is performed. For the computation repetition, generally $n(n-1)$ combination is possible in n nodes. However, repetition of computation for double critical nodes is less because the combination of double critical nodes does not include single critical nodes. Therefore, the repetition of double critical node finding is $n(n-1) - n_{cr1}(n-1)$ where n is the number of nodes and n_{cr1} is the number of single critical nodes. Similarly, the testing repetition of triple critical nodes is $n(n-1)(n-2) - n_{cr1}(n-1)(n-2) - n_{cr2}(n-2)$ where n_{cr2} is the number of combinations of double critical nodes.

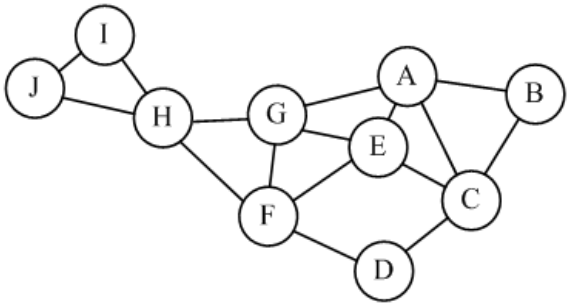


Figure 23. Example topology for multiple weak points

4.5.2 Numerical Study

100 uniformly distributed random network topologies are generated and computed for multiple critical node cases at each network density. The comparison of the average number of single, double, and triple critical nodes with 95% confidence interval in different network density is shown in Figure 24.

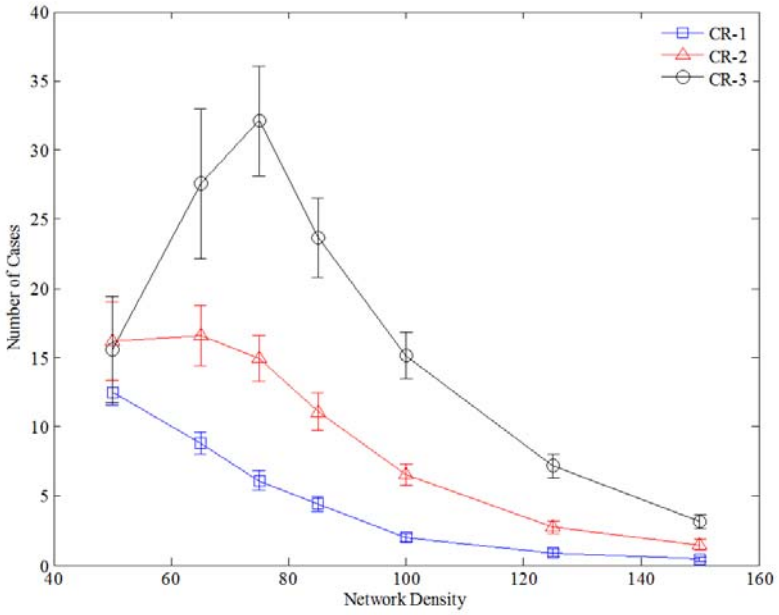


Figure 24. Comparison of single, double, and triple critical nodes

In 50 nodes network, the number of single critical nodes is less than the number of double and triple while the numbers of double and triple are close each other. As the network is denser, both the number of single critical nodes and double critical nodes decreases with a steeper slope in that of single critical nodes. The number of triple increases greatly and it reaches peak at 75 nodes network and it starts great decreasing after 85 nodes network. This phenomenon is due to the less numbers of single and double critical nodes in sparse network (i.e., 50, 75 nodes). The less number of single and double critical nodes results in the more number of remain nodes and they have more number of three nodes combinations that may fail the network. However, when the network is sufficiently dense (i.e., 85 nodes) the number of triple critical nodes decreases. At 150 nodes network, three of them are very close each other. From the observation, when the network does not hold many nodes (i.e., 50 nodes network), the number of nodes is not sufficient to form a great number of double or triple. When the network holds more

nodes such as 65 and 75, the network holds more combination of double or triple while its connectivity is not sufficient to prevent from outnumbering, especially triple case. The number of double or triple is greatly decreased when the network is dense enough to provide sufficient connectivity. Thus, when the network is not dense enough, the multiple critical nodes may become more important if multiple node failure occurs frequently, the network is attacked by multiple jammers, or combination of them.

4.6 CRITICAL POINTS AND H-HOP SUBNETWORK

Network information is not always available due to many reasons such as unreliable wireless signal or channel. In this network condition, it may be very difficult to deliver the connectivity information to the nodes in the other end of network. Then, sometimes only limited network connectivity information is available. In this chapter, we will explore how the h-hop sub-network connectivity information can be utilized to find the critical points. The critical node is considered in this chapter because critical links are already included by the nodes in homogeneous network.

4.6.1 Local Critical Points

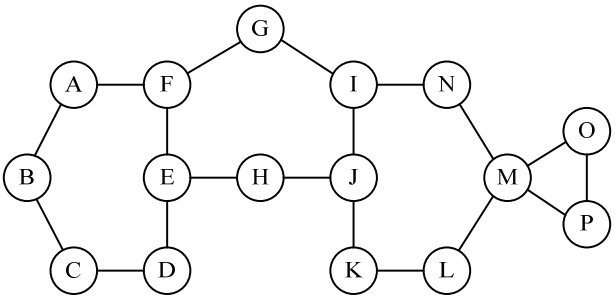
Local critical points are the critical pints in the local limited sub-network or *H*-hop sub-network. When only the limited *H*-hop sub-network connectivity information is available or it is intended to be used, each node can be evaluate itself to check if it is critical point based on obtained sub-network connectivity information. In this section, we will examine how good the local critical

points identification by h-hop sub-network information can detect global critical points found by global connectivity information.

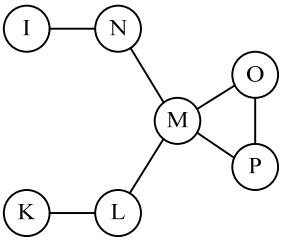
4.6.1.1 Critical point detection using *H*-hop information

Note, that we can easily adapt our proposed critical point test algorithm to utilize only local topological information as in [37]. 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 16 node network topology given in Figure 25 (a). Further consider the problem of testing whether node M or node E is critical or not using H-hop local information only. Figure 25 (b) and (c) show the *2-hop* and *3-hop* connectivity sub-networks of node M, respectively. Similarly, Figure 25 (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 the *2-hop* or *3-hop* sub-network of Figure 25 (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

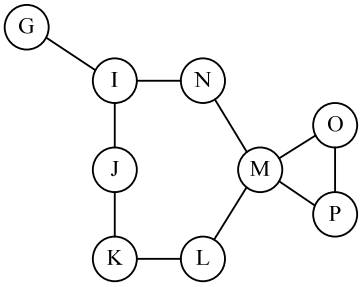
the algorithm with *H-hop* information. Hence, unlike the algorithms in [37], the critical test point algorithm will have a 100% global critical point detection rate.



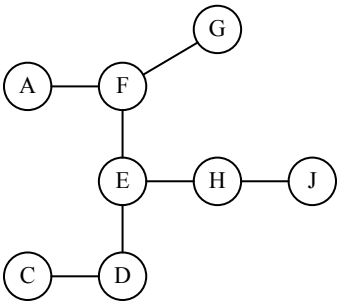
(a) 16 node network



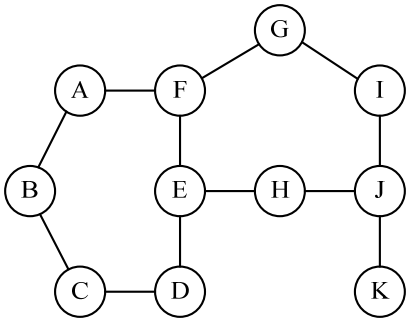
(b) 2 hop local network at Node M



(c) 3 hop local network at Node M



(e) 2 hop local network at Node E



(f) 3 hop local network at Node E

Figure 25. *H*-hop sub-networks at node M and E ($H = 2, 3$)

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 26, 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.

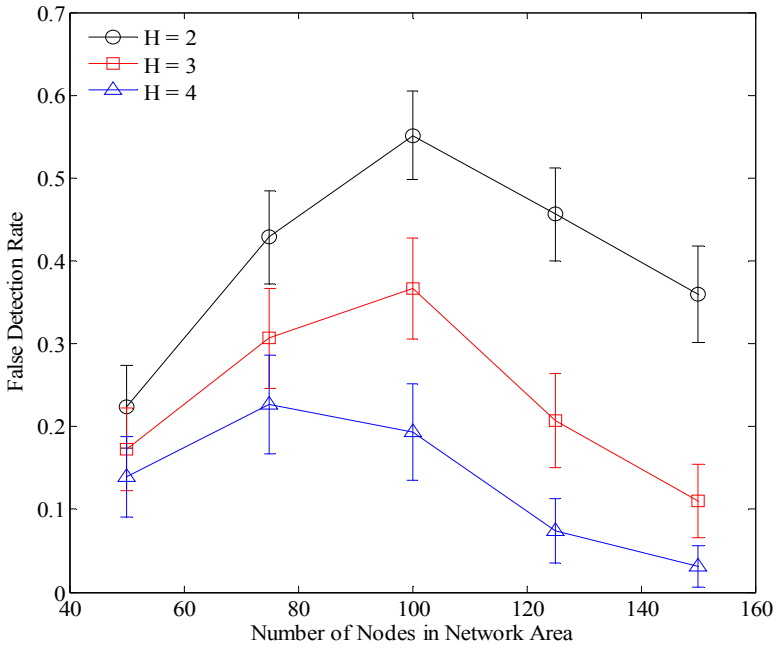


Figure 26. Single critical node False Detection rate using *H-hop* sub-networks

In local critical point identification, *H* value affects on false detection rate but it does not affect on indentifying global single critical point. However, when double critical points (i.e., the network fails when 2 points fail at the same time) is considered, *H* value is more significant and sensitive. The basic idea is that each point in double critical points is also a critical point in local

sub-network. Thus, the probability to detect double critical points using the identified local critical points increases with selected *H* value.

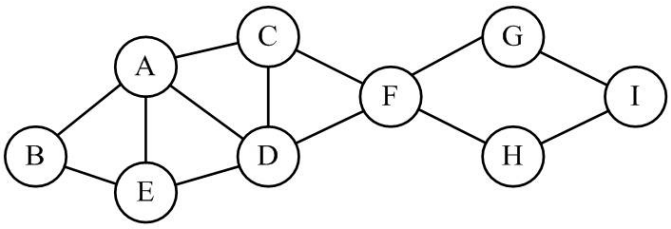


Figure 27. Sample 9 Nodes Network

4.6.2 Numerical Study

To examine how effectively the local critical node using *H-hop* information can detect the global single and double critical nodes, we compute the detection rate that is the ratio of the total number of critical node to the total number of local critical nodes of *H-hop* information for $H = \{2, 3, 4\}$. The total number critical node is the sum of the number of single and double critical nodes without repeating. For example, node F is an only single critical node (i.e., $\mathrm{CR}_1 = \{\mathrm{F}\}$) in the network as shown in Figure 27. The double critical nodes are AE, AD, CD, and GH. Then, the number of double critical nodes is 6, (i.e., the set of double critical nodes, $\mathrm{CR}_2 = \{\mathrm{A}, \mathrm{C}, \mathrm{D}, \mathrm{E}, \mathrm{G}, \mathrm{H}\}$. This results in 7 of total number of critical nodes including single and double critical nodes. Based on identified local critical nodes for selected *H* value, detection rate represents how many of identified local critical nodes fall in with a set of global critical nodes. For example, when $H = 1$, the set of local critical nodes is $\mathrm{H}_1 = \{\mathrm{F}, \mathrm{G}, \mathrm{H}, \mathrm{I}\}$ and the local critical nodes that are either single or double critical nodes are$(\mathrm{CR}_1 \cup \mathrm{CR}_2) \cap \mathrm{H}_1 = \{\mathrm{F}, \ \mathrm{G}, \mathrm{H}\}$. Then, the detection rate of single and double critical nodes for $H = 1$ is 3/7 = 0.4286.

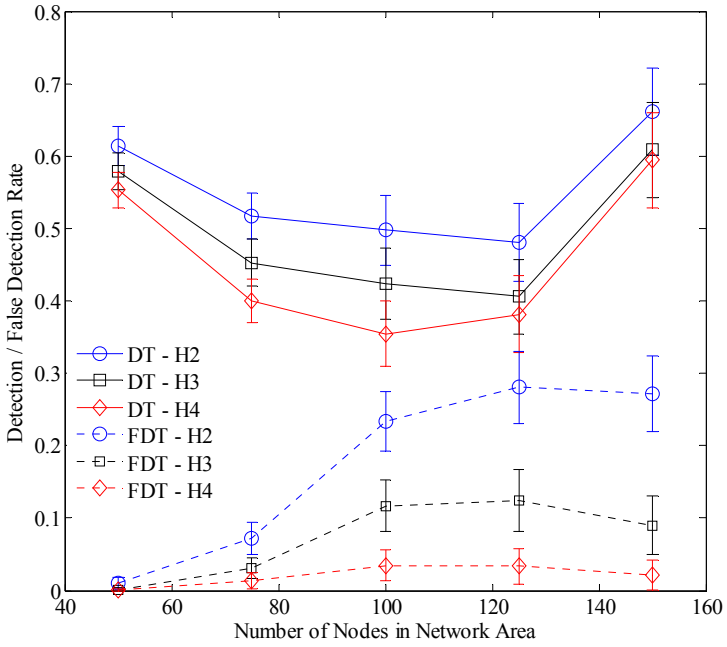


Figure 28. Single and double critical nodes Detection and False Detection rate using *H-hop* sub-networks

Same network and node conditions are employed to illustrate the effects of the local critical points on multiple critical points such as double critical points using same network topologies for each network density (100 topologies of 50, 75, 100, 125, 150 nodes in a *1500* x *1500* m^2 network area). We identified double critical nodes in each network topology and compare them with the local critical nodes for $H = \{2, 3, 4\}$. Detection and False Detection Rate of both single and double critical nodes with $H = \{2, 3, 4\}$ are computed and plotted for different network densities in Figure 28. The solid lines are detection rates and broken lines are false detection rates for each H values of 1, 2, and 3. When comparing false detection rate of both single and double critical nodes to that of single critical nodes, it decreases significantly. For example, at $H = 2$ in 75 nodes network, false detection rate decreases from 0.4283 to 0.0724 for the maximum decrement while it decreases from 0.0317 to 0.0217 at $H = 4$ in 150 nodes network for the minimum. In case of detection rate, it also decreases when single and double critical

nodes are considered comparing to that of single critical node only. In Figure 6, the lower H (i.e., $H = 2$) detects more single and double critical nodes while it also increases the false detection over all network densities.

Table 7. Difference between Detection and False Detection Rate

N	$H = 2$	$H = 3$	$H = 4$
50	0.6043	0.5777	0.5523
75	0.4451	0.4226	0.3868
100	0.2643	0.3070	0.3200
125	0.2003	0.2820	0.3483
150	0.3900	0.5189	0.5725

Note: Difference = Detection Rate – False Detection Rate

To optimize the detection of single and double critical nodes, H value has to be chosen carefully such that higher detection rate with lower false detection rate. For example, in 50 nodes network, the largest difference between both rate occurs when $H = 2$ as shown in Table 7. Then, H value of 2 provides better detection possibility. Similarly, $H = 2$ at 75 nodes network and $H = 4$ at 100, 125, and 150 nodes networks. However, this is not the only selection factor. The larger H value produces the more overheads and the longer computation time.

For the comparison of the computation time, we use 30 same connected network topologies with the network conditions such as at least one critical node, identical node capabilities, and etc. First, we compare the computation time between the critical node identification utilizing an entire topology information (i.e., single and double critical node) and H-hop local information (i.e., local critical node with $H = \{2, 3, 4\}$. Entire network topology

information uses the N x N adjacent matrix for each node while H-hop local information uses its local sub-network connectivity depending on H value. Hence, the computation time using H-hop local information is shorter than that using entire network topology information. Table 8 compares the average time computation to identify single and double critical nodes by entire network topology and H-hop local topology with H = 2, 3, 4. We measure the total time taken to identify the critical node for each topology and average them for 30 topologies as shown in Table 8. The results indicate that the computation time increases more steeply as the network is denser. For each network density, single critical node finding is greatly smaller than double critical node finding as expected. However, when the local information is used to identify the critical node, the computation time is greatly reduced. In addition, the computation time is reduced with lesser local information (i.e., smaller H) while it produces the more false detections.

Table 8. Average Computation Time for Critical Node Identification

N	**Single**	**Double**	***H* = 2**	***H* = 3**	***H* = 4**
50	0.0282	0.3476	0.0017	0.0026	0.0043
75	0.0953	2.8272	0.0051	0.0110	0.0228
100	0.2605	11.8734	0.0082	0.0234	0.0501
125	0.5041	29.9405	0.0172	0.0529	0.1187
150	0.8793	62.7270	0.0312	0.1042	0.2395

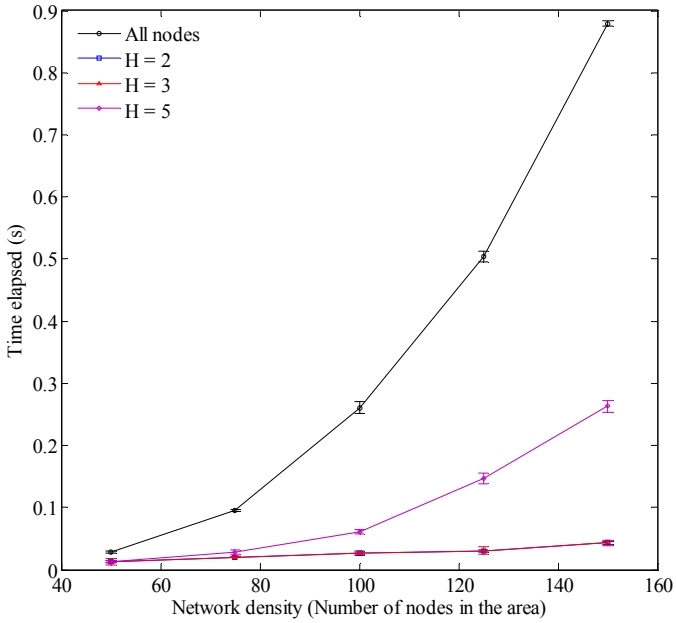


Figure 29. Time elapsed to identify critical node using different pool sets in different network density

The other advantage of using local information is to reduce the size of critical node testing set. Due to 100% detection rate for single critical node via local information, every global critical node should be in the local critical nodes set. In this process, each node tests itself for its criticality using its *H*-hop local connectivity information and reports itself as a local critical node to global testing node if it is local critical node. Once the global critical node testing node gather all local critical nodes, it tests them to identify global critical nodes. This process reduces the number of global criticality tests by half or more. Figure 29 illustrates the elapsing time to identify the critical nodes using different testing pool sets for different network densities with 95 confidence intervals. At each network density, 100 random connected topologies are tested. All nodes test examines all nodes in the network while the other tests examine the nodes only that are reported as the local critical nodes based on *H*-hop local connectivity information (i.e., $H = 2$, 3, 4). When all nodes are tested for criticality, elapsing time increases dramatically as the network is denser. However, when $H = 2$ and 3 sub-network information is used for the testing

pool, elapsing time increases slowly. Therefore, the global criticality test based on the local critical nodes in 2 or 3 of *H* value greatly reduces the computation time.

The other advantage of using local information is to reduce the size of critical node testing set. Due to 100% detection rate for single critical node via local information, every global critical node should be in the local critical nodes set. In this process, each node tests itself for its criticality using its *H*-hop local connectivity information and reports itself as a local critical node to global testing node if it is local critical node. Once the global critical node testing node gather all local critical nodes, it tests them to identify global critical nodes. This process reduces the number of global criticality tests by half or more. Figure 29 illustrates the elapsing time to identify the critical nodes using different testing pool sets for different network densities with 95 confidence intervals. At each network density, 100 random connected topologies are tested. All nodes test examines all nodes in the network while the other tests examine the nodes only that are reported as the local critical nodes based on *H*-hop local connectivity information (i.e., $H = 2$, 3, 4). When all nodes are tested for criticality, elapsing time increases dramatically as the network is denser. However, when $H = 2$ and 3 sub-network information is used for the testing pool, elapsing time increases slowly. Therefore, the global criticality test based on the local critical nodes in 2 or 3 of *H* value greatly reduces the computation time.

4.6.3 Study of *H* value

The sub-network connectivity information is based on *H* value. The larger *H* value provides lower false alarm rate and detection rate. In this section, we will examine *H* values (i.e., $H=\{ 2, 3, 4, 5, 6\}$), which may help to determine the *H* value. Simulation study is employed to illustrate how the detection and false detection rates are related in *H*.

4.6.3.1 Measurements

The measurement is required in order to evaluate the effectiveness of H value. In this section, we introduce three measurements, Detection Rate (DTR), False Alarm Rate (FAR), and Protection Rate (PTR). The Detection Rate (DTR) in this section is defined as the ratio of the total number of detected global critical nodes by local critical nodes to the total number of local critical nodes as in equation (12).

$$DTR = \frac{Number\ of\ critical\ nodes\ that\ detected\ by\ H-hop}{Total\ number\ of\ global\ critical\ nodes} \tag{12}$$

As mentioned in previous section, local critical node includes all global single critical nodes (i.e., 100% DR for single global critical nodes) where the False Alarm Rate is generated. The False Alarm Rate (FAR) is determined by the ration of the total number of local critical nodes that are not in global critical nodes to the total number of the local critical nodes as in equation (13).

$$FAR = \frac{Number\ of\ \ H-hop\ critical\ nodes\ that\ are\ not\ globally\ critical\ nodes}{Total\ number\ of\ H-hop\ critical\ nodes} \tag{13}$$

However, when double critical nodes are considered to be detected, DTR is not 100% anymore. Then, the detection rate or double critical nodes (DTR_2) is modified as it is determined the ratio of the total number of detected single and the pairs of double critical nodes to the total number of global single and double critical nodes. Similarly, FAR_2 for double critical nodes will be computed as the ratio of the total number of nodes that are not used to detect single or pairs of double critical nodes to the total number of local critical nodes. For example, consider the 9 nodes network in Figure 27 (in Section 4.6.2). This network has 1 single critical node (i.e., CR_1 = {F} and 4 pairs of double critical nodes (i.e., CR_2 = {AE, AD, CD, GH}). The local critical

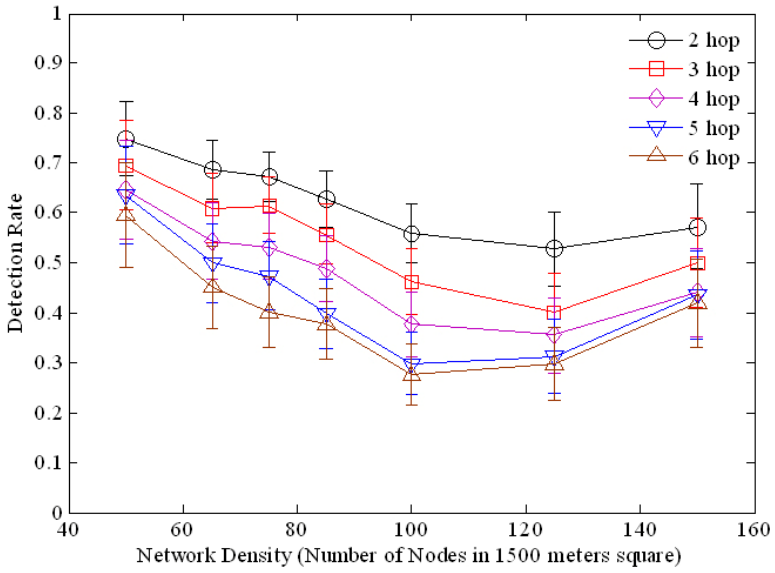
nodes by 1-hop (i.e., $H = 1$) are F, G, H and I while the local critical node by 2-hop (i.e., $H = 2$) is F only. Then, DTR for single critical node is 100% for $H = \{1, 2\}$. FAR for single critical node for $H = 1$ is determined by $^3/_4 = 0.75$ and for $H = 2$ is 0. When the double critical nodes detection rate (DTR_2) is considered, the total number of detected single and double critical nodes is 2 (i.e., F and GH) where total number of single and critical nodes is 5. Then, DTR_2 by $H = 1$ is computed by $^2/_5 = 0.4$ and FAR_2 is $^3/_4 = 0.75$. Similarly, for $H = 2$, DTR_2 is $^1/_5 = 0.2$ and FAR_2 is 0.

The last introducing measurement is Protection Rate (PTR) which mean how well the network is protected when all local critical nodes are strengthen to prevent from failure. This is same measurement to DTR but it may be different with double critical nodes. Double critical nodes are the pair of nodes that partition the network at their simultaneous failure. However, the network would not be partitioned if one of double critical nodes is strengthened and protected. Therefore, if any local critical node is one of any pair of double critical node, the double critical nodes is protected by one of local critical node. For example, the network in Figure 27 has 4 double critical nodes. The local critical nodes by $H = 1$ is F, G, H, and I. In this case, local critical node G protects the double critical nodes GH. Similarly, local critical node H also protects the network from double critical nodes GH. In this example, PR_2 is same as DR_2.

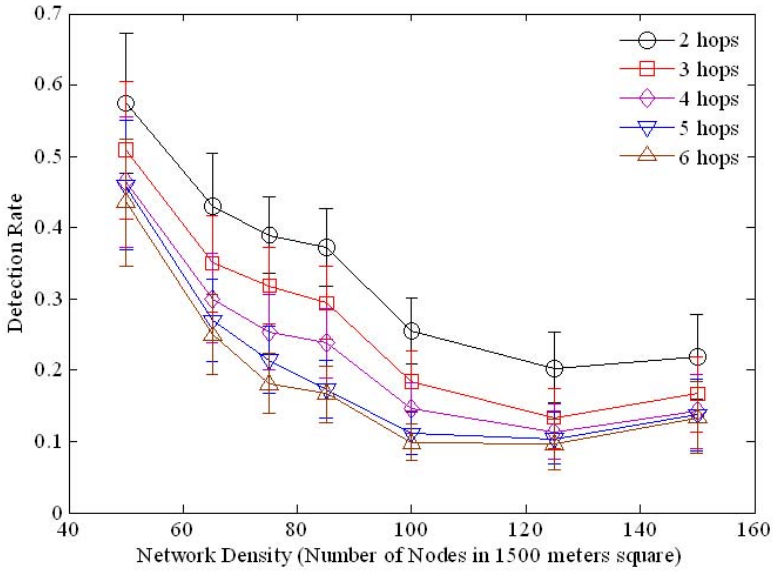
4.6.3.2 Numerical study

We use the same network and node conditions employed in Section 4.6.2 to illustrate the effects of the local critical points on multiple critical points such as double and triple critical points using same network topologies for each network density (100 topologies of 50, 75, 100, 125, 150 nodes in a *1500* x *1500* m^2 network area). We identified double critical nodes in each

network topology and compare them with the local critical nodes for H = {2, 3, 4, 5, 6}. Detection, False Alarm, and Protection Rate of single, double, and triple critical nodes with H = {2, 3, 4, 5, 6} are computed and plotted for different network densities. The Detection Rate of single and double critical nodes (DTR_2) is shown in Figure 30(a). DTR_2 represents how the H-hop of local connectivity information detects the single and double critical nodes over different network densities. DTR_2 decreases with H value in all network densities. The DTR2 decreases as the network is getting denser while it slightly increases as network is denser than 100 nodes network. The local critical nodes detect more correctly in sparse network. DTR_3 is plotted in Figure 30(b) where DTR3 includes single, double, and triple critical nodes. It also shows better detection rate with smaller H and worse detection rate in denser network.



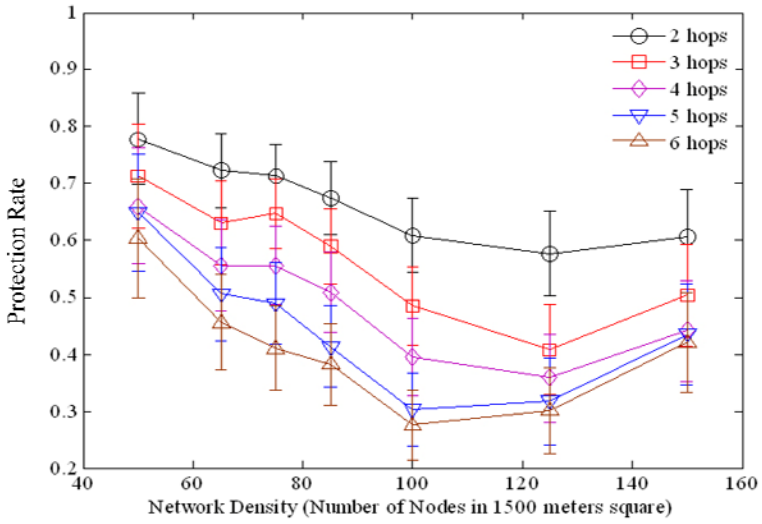
(a) Detection Rate of single and double critical nodes (DTR_2)



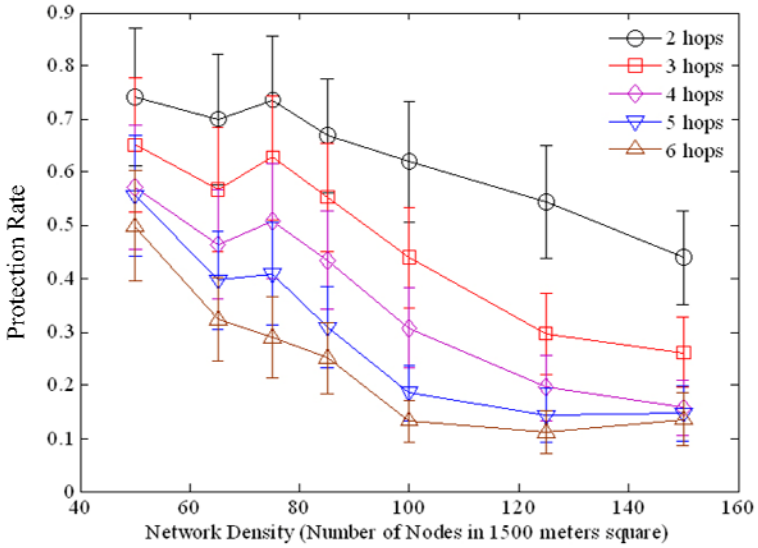
(b) Detection Rate of single, double, and triple critical nodes (DTR_3)

Figure 30. Detection Rates (DTR)

PTR_2 does not show significant improvement when it is compared to DTR_2 in overall. It is slightly higher protection rate when the network is denser (i.e., 100 and 125 nodes network) at $H = 2$. However, PTR_3 improves significantly at H = {2, 3, 4}.

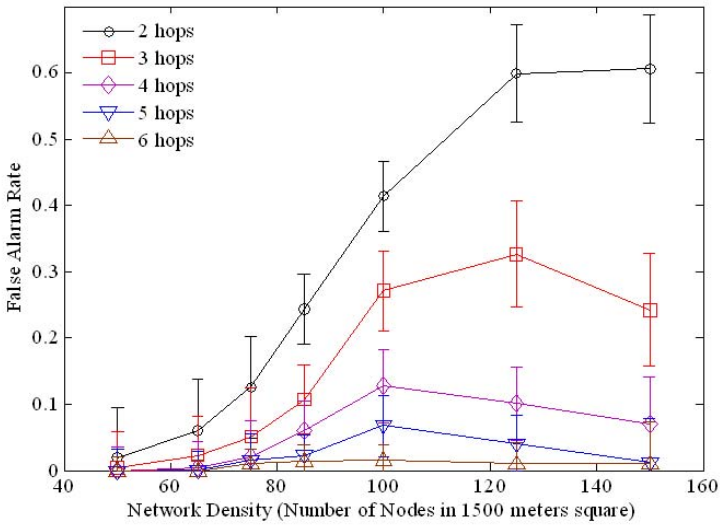


(a) Protection Rate of single and double critical nodes (PTR_2)

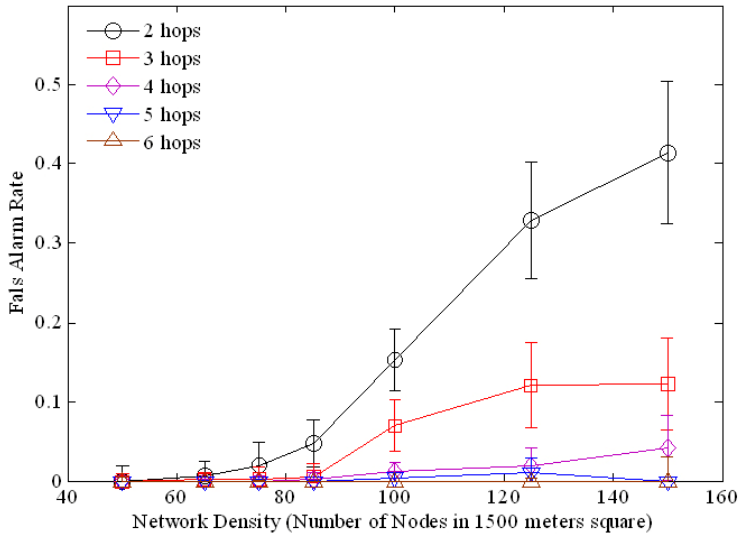


(b) Protection Rate of single, double, and triple critical nodes (PTR_3)

Figure 31. Protection Rates (PTR)



(a) False Alarm Rate of single and double critical nodes (FAR_2)



(b) False Alarm Rate of single, double, and triple critical nodes (FAR_3)

Figure 32. False Alarm Rates (FAR)

The False Alarm Rates of single and double critical nodes (i.e., FAR_2) are illustrated in Figure 32(a), which indicate the percentage of local critical nodes are not valid in detection or protection of single and double critical nodes. Similarly, FAR_3, False Alarm Rates of single, double, and triple critical nodes, are shown in Figure 32(b). False alarm rate increases when H is smaller (i.e., H =2). Local critical nodes detect more correctly as the larger combinations of the critical nodes (i.e., single, double, and triple critical nodes). FAR_3 is significantly smaller than FAR_2 in all network densities and H values. The local critical nodes by larger H value are more likely single critical nodes because the local information with larger value of H becomes global. Therefore, the FAR with larger H value is close to 0.

4.7 DISCUSSIONS

The critical points such as articulation nodes or bridge links threat the network and the failure of any one of those points partitions the network. In this chapter, two heuristic algorithms are proposed, which are effectively identifies the critical points than other metrics. Algorithm I and II identifies all critical points but Algorithm II cannot aware the existing network partition. The time complexity of Algorithm I is very high, but it provides much more information that can be used in further proposing resilient schemes in this paper. Algorithm I can also be used to identify the critical nodes and critical links. Using critical point identification algorithm, we study the critical nodes. Based on our findings, those critical points are not predictable in terms of their number or positions in the network. This indicates that the resilient scheme should protect the critical points directly because of the uncertainty. Algorithm I requires the global network topology information. However, global network topology cannot be obtained sometimes. In this case, local subnetwork information could be used to predict the possible global critical points. We study the size of local subnetwork in terms of H and all local critical points are also critical points in global. However, it also causes the false detection. As the H value increases, the false alarm rate decreases. The more local critical points are found in small H value, which may create additional unnecessary protection actions. Therefore, the H value is selective to the condition of the network and resilient scheme.

4.8 CONCLUSTIONS

In this chapter, we proposed two heuristic algorithms to identify the critical points. Algorithm I provides more information of clusters, which is used for resilient schemes in this paper. The critical links can be also found by Algorithm I. The number of critical points and their positions in the network are random and unpredictable. Therefore, we need to protect each critical point. In the rest of this paper, we propose the resilient schemes that protect the critical points in order for the network to be more reliable in any failures.

5.0 CONNECTIVITY IMPROVEMENT SCHEMES IN HOMOGENEOUS WIRELESS NETWORK

In previous chapter, we study about the critical points such as how to identify the critical points (i.e., nodes and links), behaviors, positions, and etc. From our study, it is known that the critical points are randomly locating on the network topology. Therefore, the pre-determined protection of the critical points is not easy in MANETs because the locations of the critical points are unpredictable. In this chapter, we assume that the network is homogeneous whose nodes have identical node properties. Then, we propose two critical points protection schemes that improve the network connectivity in homogeneous wireless network once the nodes are deployed. One is localized and the other is globalized scheme.

5.1 LOCAL RESILIENCE SCHEMES

The first scheme we propose is localized resilience scheme to improve the connectivity of the homogeneous wireless network. This scheme is to protect the identified critical points in order to eliminate the risk from the network partitioning due to failure of critical points. The network topology has k-node connectivity it protects the network from any combination of single, double, triple, … up to (*k-1*) node failures. Therefore, the network is *k*-connected if the network does not have any critical nodes up to ($k - 1$) multiple node combinations cases. We propose a critical

node management approach to providing *k*-connectivity. Specifically one uses the Algorithm I 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 this chapter 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) and (2) 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 technique establishes 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.

5.1.1 Local Full Mesh (LFM)

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. At each critical node, it checks the direct link connectivity between each pair of neighbor nodes in neighbor nodes set **B** (i.e., $\mathbf{B} = \{B_i; i = 1, 2, 3, \ldots, d\}$ where *d* is node degree). Table 9 illustrates the LFM algorithm.

Table 9. Algorithm of the Local Full Mesh (LFM) Scheme

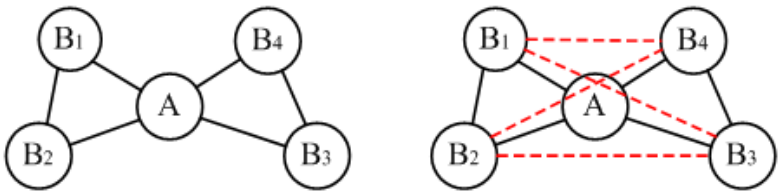
```
% A is an adjacent matrix of the network
%     (A(i,j)= 1 if direct link exists, otherwise a_ij = 0)
% d_i is a node degree of node i
% l_crn is a number of critical nodes
% Crn is a set of critical nodes
% B_i(·) is a set of neighbor nodes of node i
% l_px is a number nodes that its transmission power needs to be increased
% N_px is a set of nodes that its transmission power needs to be increased
% P_x(N_px(l_px)) Increased amount of transmission power of node N_px(l_px)

% Inputs: A, d_i, N_ngb(·), l_cr, Cr
% Outputs: l_px, N_px, P_x(·)

begin
   l_px = 0;
   for i = 1 to l_crn
      for j = 1 to d_Crn(i)
         for k1 = 1 to B_Crn(i)(j)
            for k2 = k1 + 1 to B_Crn(i)(j)
               if A(k1, k2) = 0
                  increase T_x power until A(k1, k2) = 1;
                  l_px = l_px + 1;
                  set new T_x power into P_x(N_px(l_px));
               endif
            endfor
         endfor
      endfor
   endfor
end
```

For example, in Figure 33(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 pairs of neighbor nodes are to be examined for their direct link connectivity (i.e., $A(B_1,B_2) = A(B_3,B_4) = 1$, $A(B_1,B3) = A(B_1,B4) = A(B_2,B_3) = A(B_2,B_4) = 0$). Then, all nodes in pairs of neighbor nodes who do not have direct link (i.e., B_1B3, B_1B4, B_2B_3, and B2B4) 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 33(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. The number of nodes that increase their transmission power is 4 and 4 new links are established around the critical node A in this example.



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

Figure 33. Local Full Mesh scheme around critical node A

5.1.2 Least Number of Link with Least Cost (LNLLC)

The Least Number of Link with Least Cost (LNLLC) 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 LNLLC 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. Next, it searches the all possible link combinations. The possible link is the link that can be established between separated clusters. This link can be determined the least cost one among all the possible pairs of nodes from each cluster. Based on found set of possible links, it solves the optimization problem to find the link combination among possible links set that has the minimum total cost and prevents the local sub-network partition when the critical node is unavailable. In general, the minimum number of links to connect n nodes is $n - 1$. Then, the number of possible links combination is $N_{lcl} - 1$ where N_{lcl} is the number of local clusters.

Table 10. Algorithm of the Least Number of Links with Lest Cost (LNLLC) Scheme

% ***Import*** Variables from *LFM* Algorithm

% N_{lcl} is a number of local clusters
% ***mLC*** is a minimum link cost matrix between local clusters
% ***CL_A*** is a local cluster adjacent matrix
% ***PLS*** is a possible links set
% ***AL_A*** is a minimum cost local connectivity matrix between clusters
% ***L(·)*** – Laplacian Matrix
% ***Eig₂(·)*** – Second smallest eigenvalue

% **Inputs:** *A*, $N_{ngb}(\cdot)$, l_{cr}, *Cr*, *mLC*
% **Outputs:** *AL_A*

begin

for $i = 1$ to l_{cr}

obtain local clusters based on $\boldsymbol{B_{Crn(i)}}$;
remove *Crn*(*i*); ***get PLS***;
set *CL_A* = zeros(N_{lcl});

for all ($\boldsymbol{N_{lcl} - 1}$) of links combination between clusters from ***PLS***

Create Connectivity in *CL_A*;
If (Eig_2 (*L* (*CL_A*) $\neq$ 0) $\wedge$ (***sum***(***sum***(*CL_A* × *mLC*)) is minimum)
AL_A = *CL_A*;
end if

end for
end for
end

Solve: $\mathbf{min}\left(\sum_{i=1}^{N_{lcl}}\sum_{j=1}^{N_{lcl}}(CL\_A \times mLC)\right) \wedge Eig_2\left(L(CL\_A)\right) \neq 0$

Conditions:

c1: Possible links set **PLS** = {minimum cost links between clusters}

c2: $N_l = N_{lcl} - 1$

where

PLS – Possible links set between clusters

CL_A – Local connectivity matrix between clusters representing testing combination of the links from **PLS**

mLC – local cost matrix between clusters

N_{lcl} – Number of local partitioned clusters

N_l – Number of testing links combination from **PLS**

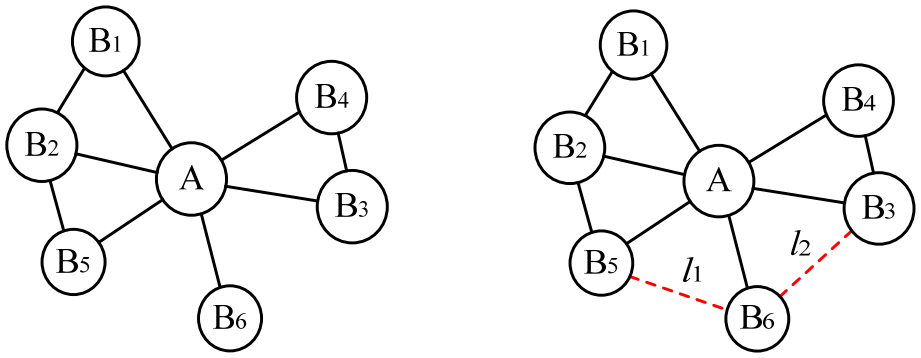
$L(\cdot)$ – Laplacina matrix

$Eig_2(\cdot)$ – Second smallest eigenvalue

This process is performed on each critical node to find the minimum cost links set that protects that critical node. The LNLLC algorithm is shown in Table 10.

For example, in Figure 34(a), node A is a critical node and the number of clusters when node A fails is three ($N_{lcl} = 3$). Then, the minimum number of additional links to relax the single point of failure is two ($N_{lcl} - 1 = 2$). Then, we consider Least Number of Links with Least Cost (*LNLLC*) scheme to protect this network with the distance for the cost metric. If the node position is known and link cost is only based on distance between nodes, it is possible to select the least cost links using *LNLLC*. 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 9(b). 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 [36, 37]. 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.



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

Figure 34. Additional link selection in LNLLC schemes

5.1.3 Implementations

In this session, we provide how above proposing resilient techniques can be implemented. Both techniques require 1-hop connectivity at each critical node, which can be obtained using periodically packets such as “Hello” packet. All neighbor nodes of each critical node send their 1-hop connectivity information to critical node. Once the critical node gathers all its neighbor nodes’ 1-hop connectivity information, it can identify the pair(s) of neighbor nodes needed to be connected for each technique. In LFM, each critical node sends each pair of unconnected neighbor nodes to create local full mesh. At neighbor node who receives the target node to connect, it increases its transmission power by a prefixed increment, ΔTx, periodically until it can communicate with assigned target node. In LNLLC, critical node knows position information of its neighbor nodes and it determines the minimum number of pair of neighbor

nodes. It can also send determined pair of neighbor nodes with distance information to be connected. Then, the time to connect assigned pair of nodes is least.

These techniques can be also applied on the local critical nodes that are identified by the subnetwork (i.e., H = 2, 3, 4, …). Those critical node findings produce the false alarms but they can also be used to protect the multiple critical nodes cases as discussed in Chapter 4.6

5.1.4 Numerical Study

The proposing above two resilient techniques are applied on identified critical nodes in order to make it at least 2-connected network. In session, we also implement those techniques on critical nodes identified by global and subnetwork information (H = 2, 3, 4).

5.1.4.1 Local resilient techniques by global network information

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 $1500 \times 1500\ m^2$. 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. 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 and 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 [24,28].

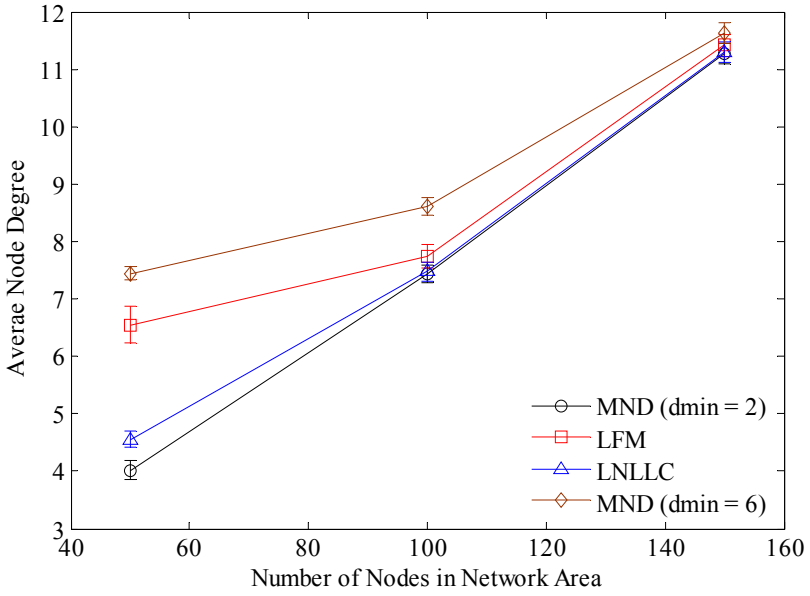
First we examine the effectivness of the proposed schemes in providing k = 2 connectivity for the entire network. Each topology possesses at least one single critical node (i.e., 1-connected). Table 11 shows the percentages of 2-connected networks for each of the schemes for network densities of 50, 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, 3, 4, 5, 6) is met. The proposed schemes LFM 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 and minimum number of node degree, whereas the proposed LFM and LNLLC schemes always result in a 2-connected network. The greater number of minimum node degree provides the better connectivity for all node densities. At d_{min} = 2, the probability of 2-connected network is 6.67%. This probability increases up to 36.67% at d_{min} = 3, 80% at d_{min} = 4, and 100% at d_{min} = 5. Minimum node degree algorithm does not guarantee 2-connectivity up to d_{min} = 5 (i.e., P(2-connected) = 93.33% at 100 nodes density). When minimum node degree is set to 6, all networks become 2-connnected in random topology. Yet, the random topology still has a chance to have one or more critical points even with the larger minimum node degree condition.

Table 11. Connectivity Percentages over 30 Topologies for k = 2

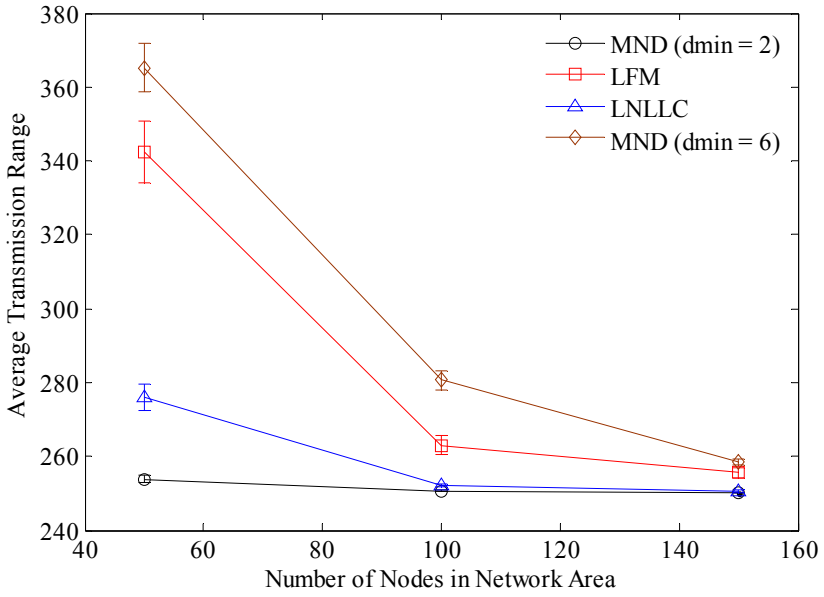
N	No Protection	MND ($d_{\min}$ = 2)	MND ($d_{\min}$ = 3)	MND ($d_{\min}$ = 4)	MND ($d_{\min}$ = 5)	MND ($d_{\min}$ = 6)	LFM	LNLLC
50	0	0.0667	0.3667	0.8	1.0	1.0	1.0	1.0
100	0	0.3667	0.6667	0.8333	0.9333	1.0	1.0	1.0
150	0	0.4333	0.7667	0.9	100%	1.0	1.0	1.0

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, 3, 4, 5, 6)

LFM and LNLLC provides the 100% guarantee to 2-connectivity. However, there are some tradeoffs between LFM and LNLLC. Illustrating tradeoffs, we compare the LFM and LNLLC with MND. We select d_{min} =2 and 6 (i.e., MND(2), MND(6)) because 2 is the desired connectivity based on minimum node degree assumption and probability of 2-connectivity reaches 100% at MND(6) form our results. For the comparison, 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 35(a) and (b), respectively. The average node degree provides a metric of connectivity and interference. In all nodes densities, MND(6) has the highest average node degree since it creates at least 6 neighbor nodes for all nodes. LFM has the second highest average node degree (i.e., MND(2) 4.02, LNLLC 4.55, LFM 6.55, MND(6) 7.44) in the sparse network case (i.e., 50 nodes network). 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.



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



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

Figure 35. Average Node Degree, and Transmission Range at $k = 2$

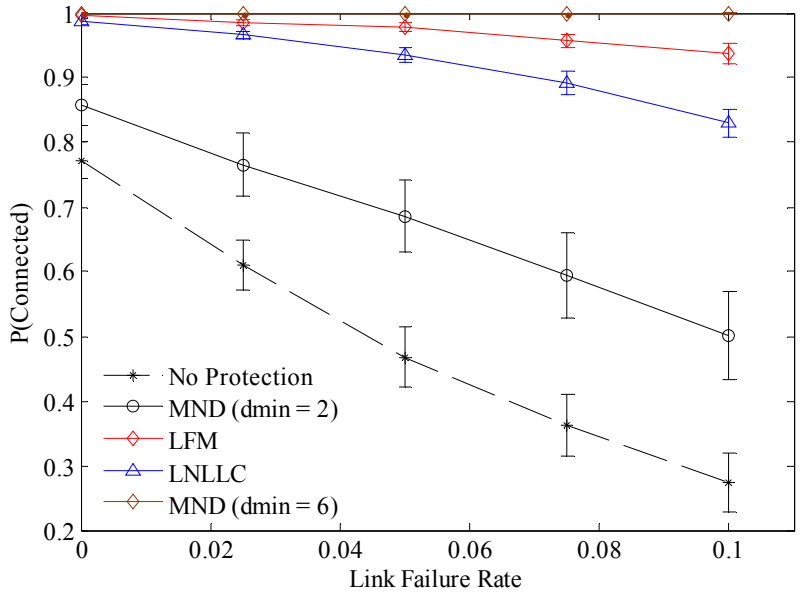
The average transmission range can be related to the average energy consumption of the network since increasing the transmission range is achieved by increasing the transmission power of the node. As shown in Figure 35(b), MND(6) requires significantly more energy than the other schemes for the 50 nodes network (i.e., 365.22 m) case since it requires a larger range to maintain minimum node degree of 6. LFM follows second largest energy consumption than others for the 50 nodes network (i.e., 342.46 m). As network density increases, MND(6) and LFM still consume more than others but it is not as significantly large as it is in 50 nodes network. LNLLC scheme consumes almost about the same energy as MND(2) does but the LNLLC scheme provides full resilience to any single node failure unlike MND(2).

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

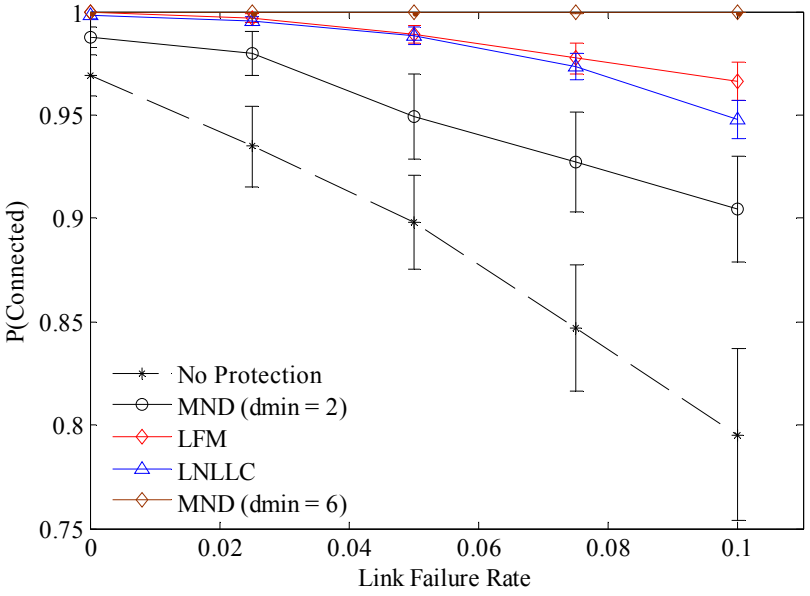
node failure was set to result in an average of one node failure for each network density (i.e., P_{nf} = *1/N*). The link failure rate was varied (P_{lf} = *0.00*, *0.025*, *0.05*, *0.1*), where P_{lf} = 0.1 means that on average 100 x P_{lf} = 10 % of the links fail in the network. For each of the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

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 36(a), (b), and (c), respectively. As one would expect LFM improves *P*(*Connected*) the most in our proposed schemes. 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(2) improves it to 0.9543 for the minimum improvement.

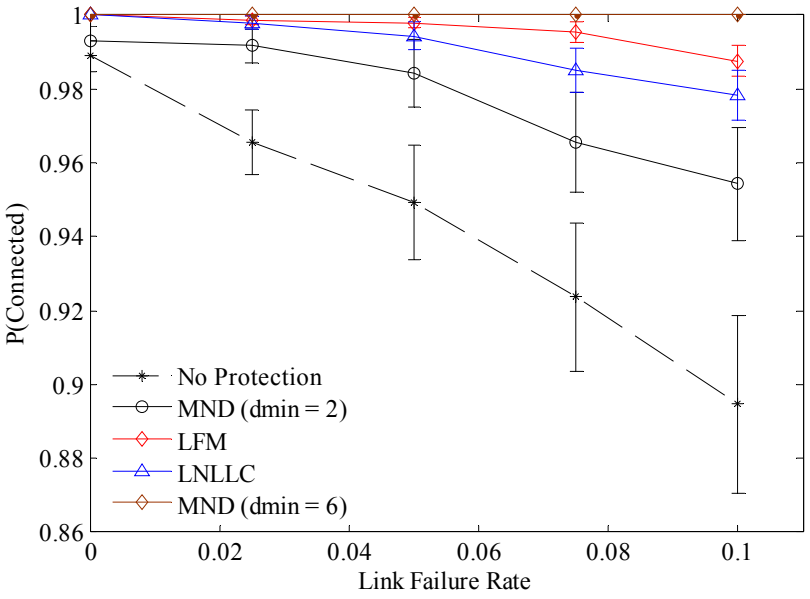
When the two proposed resilient schemes are compared with the Minimum Node Degree techniques (i.e., MND(2) and MND(6)), all of them improve the probability of network connectivity more than MND(2) at any network density, but not for MND(6). MND(6) improves the probability of the network connectivity up to almost 100% for all failure cases in all network densities. Another observation is that *P*(*Connected*) decreases faster with MND(2) 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(2) while it decreases from 0.9877 to 0.8297 with LNLLC (LNLLC has the largest decrease in *P*(*Connected*) in two 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 $P_{nf} = 0.0067$

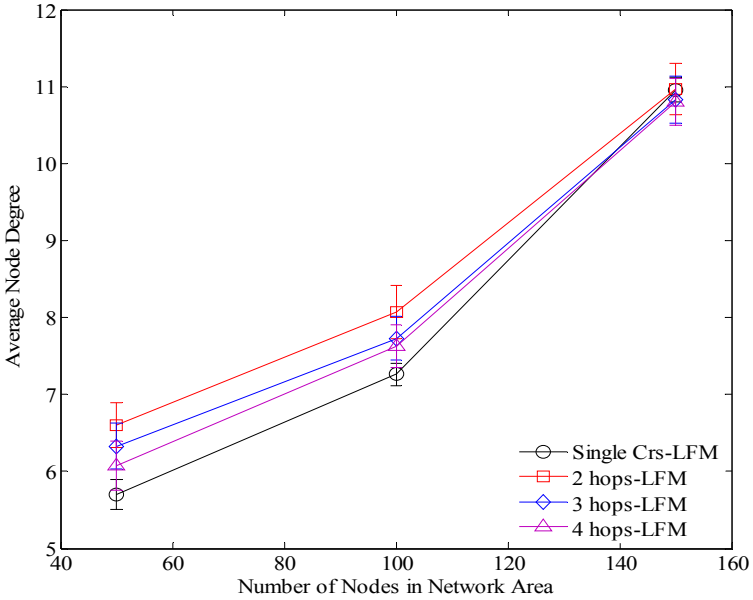
Figure 36. Probability of Network being connected with 95% CI utilizing Minimum Node Degree, Local Full Mesh (LFM), Least Number of Link with Least Cost (LNLLC) in (a) 50 node with $P_{nf} = 0.02$, (b) 100 node with $P_{nf} = 0.01$, (c) 150 node network with $P_{nf} = 0.0067$

In 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 LNLLC 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 $P_{lf} = 0.1$.

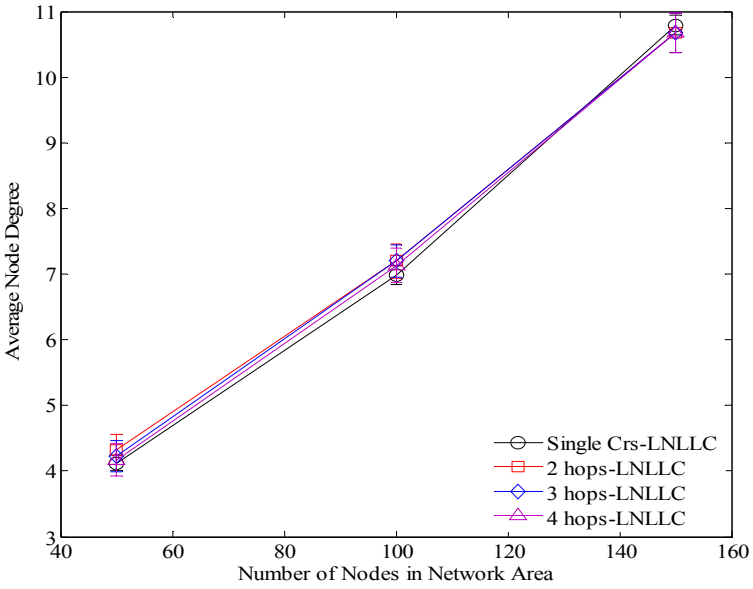
5.1.4.2 Local resilient techniques by local network information

Now we apply our schemes to the critical nodes found by H-hop subnetwork with $H = \{2, 3, 4\}$. Since all single critical nodes are identified by H-hop subnetwork (i.e., $H = 2, 3, 4$) with false detection, all 30 random 1-connected network topologies become 2-connected with deploying our schemes on local critical node in all network density. However, H-hop approach

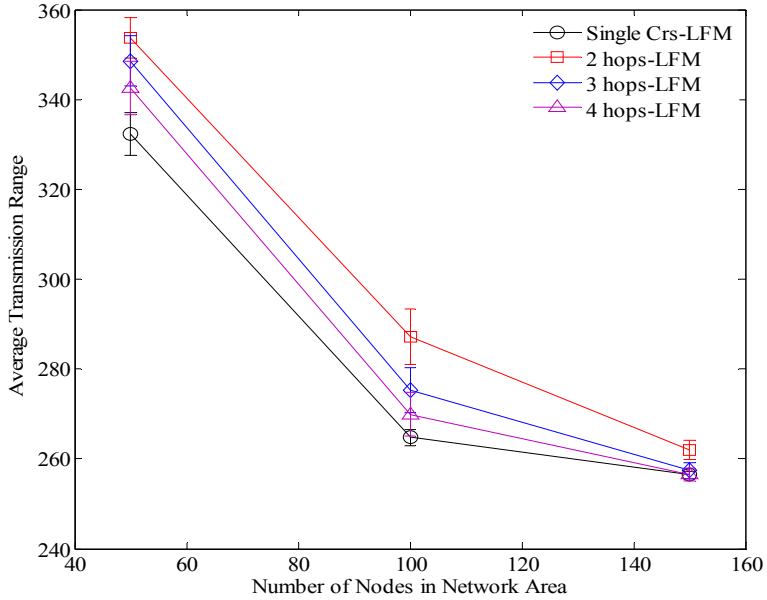
may generate the additional unnecessary transmission range increments and it also induces the additional interference or energy consumption for 2-connected network due to its false. The average node degrees and average transmission ranges in *H*-hop subnetwork approach with our schemes are illustrated in Figure 36. Average node degrees of LFM and LNLLC are showed in Figure 37(a) and (b), respectively and average transmission ranges for LFM and LNLLC are in (c) and (d), respectively. With LFM, the average node degrees and the average transmission ranges are largely induced with smaller *H* in sparse network (i.e., 50 and 100 nodes). However, with LNLLC, average node degrees are almost similar for all cases while average transmission ranges show differences but it is relatively small comparing to that with LFM (i.e., 332.33m with global and 353.71m with $H = 2$ local in LFM and 278.32m with global and 285.71m with $H = 2$ local in LNLLC at 50 network). 100 nodes network produces the great difference of average transmission range between two schemes (i.e., 264.77m and 287.20m with LFM and 252.39m and 255.76m with LNLLC). Consequently, LNLLC scheme on local critical node produces less interference and energy consumptions than LFM. Furthermore, *H*-hop subnetwork approach improves the network resilience more because local critical node may not be global single critical node but one of multiple critical nodes (i.e., double, triple) as described in Chapter 4.



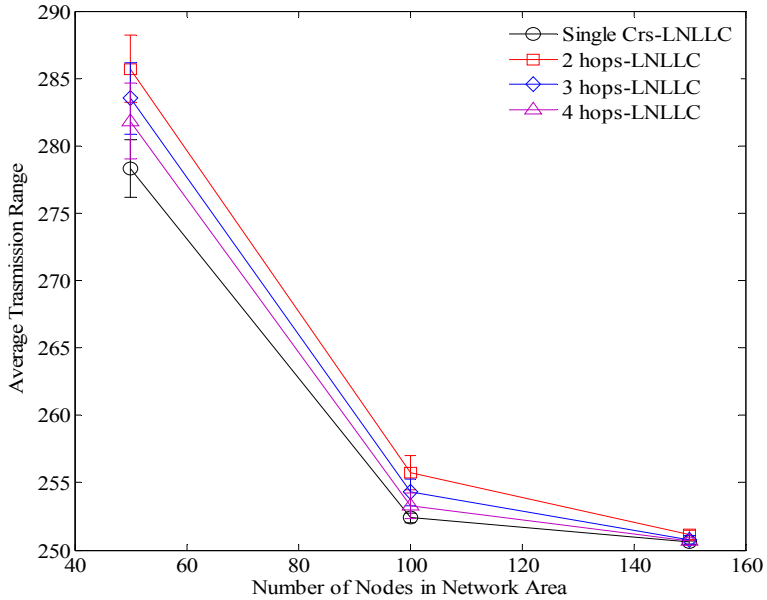
(a) Average Node Degree of LFM at $k = 2$



(b) Average Node Degree of LNLLC at $k = 2$



(c) Average Transmission Range of LFM at $k = 2$



(d) Average Transmission Range of LNLLC at $k = 2$

Figure 37. Average Node Degree and Average Transmission Range of LFM and LNLLC at $k = 2$ for $H = 2, 3, 4$

5.2 GLOBAL RESILIENT SCHEMES

We propose local resilient scheme that protects the network from partitioning due to critical point failure. The local resilient scheme control the network topology locally (i.e., neighbor nodes around the critical nodes). In this section, we propose another resilient scheme that controls the topology globally. This scheme identifies the partitioning clusters when the critical point fails and finds the links between clusters. This scheme utilizes the information gathered from the heuristic critical point identifying algorithm I (i.e., eigenvalues and eigenvectors of Laplacian matrix). The idea of this scheme is simply merging the partitioned clusters based on the interesting metrics. For the first of all, we classify the critical points into 3 types of critical points, which may help to greatly reduce the computation. Then, it finds the minimum number of additional links to protect the critical points by cluster based approach.

5.2.1 Critical Points Classifications

The critical points in general consist of critical nodes and critical links (i.e., articulation node and bridge link). Each critical point partitions the network when it fails. Then, the resilient scheme for the critical points needs to be done for each point, which simple identifies the additional link(s) for each absence of critical point. However, when the global information is available, it can find the overlapping cost optimal additional links that protects the critical point and do not contain any critical node. In other words, a set of additional links that do not contain any critical node can be used to protect one or more critical points. This advantage motivates us to classify critical points. We classify the critical points into 3 types; Critical link (***Crl***), Critical node (***Crn***), and Combined Critical Links (***CCrl***). Critical link is simply a bridge link and critical node is an

articulation node. The combined critical links is the consecutively connected critical links that share the cost optimal additional protection link(s).

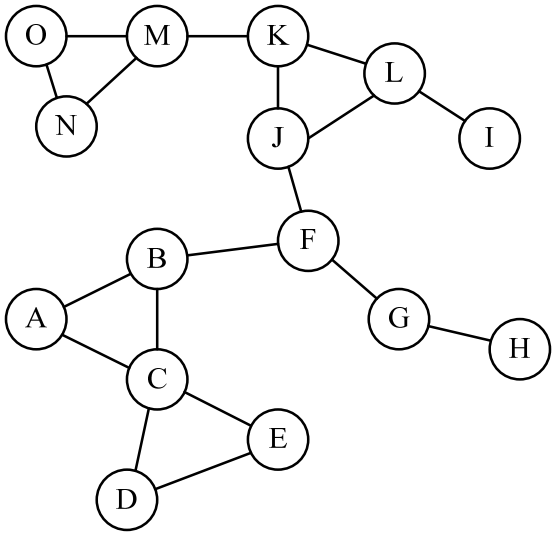


Figure 38. 15 nodes network that has several critical points

For example, Figure 38 illustrates the 15 nodes network that has several critical points; 1 critical node (i.e., ***Crn***= {C}) and 6 critical links (i.e., ***Crl*** = { l_{BF}, l_{FG}, l_{GH}, l_{JF}, l_{IL}, l_{KM}}). In our critical point classification, however, those critical points become 1 critical node (i.e., ***Crn*** = {C}), 2 critical links (i.e., ***Crl*** = {l_{IL}, l_{KM}}, and 1 combined critical links (i.e. ***CCrl*** = {$l_{BF}l_{FG}l_{GH}l_{JF}$}). In this example, we use the link cost matrix for the creation of the additional link between pair of nodes as shown in Table 12. In this scheme, we exclude all the critical points from additional link selection. Then, we find the least number of additional links. When the link between F and G or G and H is removed, node H is isolated and it needs to find minimum cost additional link to other nodes in the rest of the network. It uses the cost matrix in Table 12 to find one near, which is node I. When link between B and F or F and J is removed, the minimum cost additional link is l_{AN} by the same procedure. Those same additional links l_{AN} and l_{HI} are also selected when the links in combined critical links (i.e. ***CCrl*** = {$l_{BF}l_{FG}l_{GH}l_{JF}$}) are not presenting.

The four connected critical links find additional links that are same results from each four critical link. Then, the computation time is reduced by once than four times. Note that the cost matrix weighted by link cost. The link cost in wireless is generally related to the receiving signal strength where the signal strength attenuates mostly by the distance. The weaker receiving signal strength may cause several more transmissions to receive the correct packet compare to the one with stronger one. Therefore, the link cost may not change significantly if the network topology stays same.

Table 12. Cost Matrix to create the additional link between nodes based on distance

	A	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**J**	**K**	**L**	**M**	**N**	**O**
A	0	0	0	8	11.6	12.2	16.8	23	23.1	13.4	17.2	20.5	13.5	12.1	7.2
B	0	0	0	10	8.5	0	10.5	17.5	16.5	6	10.5	13.1	9	5.5	10.5
C	0	0	0	0	0	8	10.2	17	19	11.5	16.8	18	16	11.5	17
D	8	10	0	0	0	14	14.5	19.8	24.2	18.2	23	24	22	17.5	22.1
E	11.6	8.5	0	0	0	7.8	6	11	17	13.2	19	18.2	19.8	17.6	22
F	12.2	0	8	14	7.8	0	0	19.6	8	0	8	7	11.8	11.1	15.5
G	16.8	10.5	10.2	14.5	6	0	0	0	7.9	8.7	13.5	10	17.1	17.9	21.5
H	23	17.5	17	19.8	11	19.6	0	0	8.5	19.2	18.5	13.4	22.8	23.9	28
I	23.1	16.5	19	24.2	17	8	7.9	8.5	0	8.5	9.7	0	16	19.8	21.9
J	13.4	6	11.5	18.2	13.2	0	8.7	19.2	8.5	0	0	0	5	8.3	11
K	17.2	10.5	16.8	23	19	8	13.5	18.5	9.7	0	0	0	0	9.3	10
L	20.5	13.1	18	24	18.2	7	10	13.4	0	0	0	0	9.8	15	16.5
M	13.5	9	16	22	19.8	11.8	17.1	22.5	16	5	0	9.8	0	0	0
N	12.1	5.5	11.5	17.5	17.6	11.1	17.9	23.9	19.8	8.3	9.3	15	0	0	0
O	7.2	10.5	17	22.1	22	15.5	21.5	28	21.9	11	10	16.5	0	0	0

Once the critical points are classified, it discovers the set of nodes for each critical point. To determine the additional link(s) for the critical point, we remove the set of nodes associated with this critical point because those end nodes of the critical links are critical as well; either node failure causes the network partition. For example in above 15 nodes network, considering

the combined critical links, removal of all the associated nodes such as {B, F, G, J} divides the network into 3 clusters. Then, the additional links are determined to reconnect those separated clusters. Therefore, we use the set of associated nodes for critical link and combined critical link.

5.2.2 Cluster Based Merging Schemes

Here, we propose the globalized resilient scheme (i.e., Cluster Based Merging Scheme (*CBMS*)). *CBMS* firstly assorts the partitioning clusters and the set of their member nodes caused by absence of critical point, which results in adjacent matrix of the cluster. Then, it computes the minimum cost additional link(s) between clusters by comparing the pairs of nodes from each cluster. From this set of possible additional link, it finds the set of links that connects the clusters with minimum total cost.

5.2.2.1 Cluster adjacent matrix

CBMS approach is based on identifying the isolated groups (i.e. clusters) of nodes and determining the links required to connect between clusters. Consider a network partitioned into N_{CL} clusters. The Laplacian matrix L with re-labeled node IDs can be written as

$$L = \begin{pmatrix} L_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & L_{NCL} \end{pmatrix} \tag{14}$$

where each L_k represents the Laplacian matrix of a connected cluster C_k of nodes. There are N_{CL} zeros in the eigenvalue set $\boldsymbol{\lambda} = [\lambda_1, \lambda_2, \ldots \lambda_N]$ of $L(t)$. From the N_{CL} eigenvectors x_i associated with the zero eigenvalues one can determine the nodes in each cluster. Specifically the nodes in a cluster will have non-zero values in the eigenvector associated with a zero eigenvalue [43]. In

eigenvector, the row represents the node ID. For example, consider 6 nodes network as shown in Figure 39.

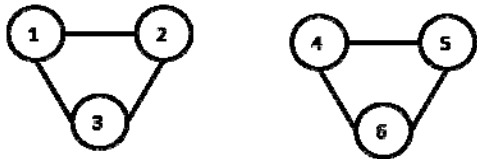


Figure 39. Sample 6 nodes network

$$A = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix} \qquad L = \begin{bmatrix} 2 & -1 & -1 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 & 0 \\ -1 & -1 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & -1 & -1 \\ 0 & 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & -1 & -1 & 2 \end{bmatrix}$$

The adjacent matrix and its Lpalcian matrix of this network are shown above. Then, the corresponding eigenvalues are found as $\boldsymbol{\lambda}$ = [0, 0, 3, 3, 3, 3] and its eigenvectors are shown below.

$$\mathbf{X} = \begin{bmatrix} 0.5774 & 0 & 0 & 0.2673 & 0 & 0.7715 \\ 0.5774 & 0 & 0 & -0.8018 & 0 & -0.1543 \\ 0.5774 & 0 & 0 & 0.5345 & 0 & -0.6172 \\ 0 & 0.5774 & 0.7634 & 0 & 0.2895 & 0 \\ 0 & 0.5774 & -0.6325 & 0 & 0.5164 & 0 \\ 0 & 0.5774 & -0.1310 & 0 & -0.8059 & 0 \end{bmatrix}$$

Since there are two zeros of eigenvalues, the number separated clusters are two. We are interested in the eigenvector whose eigenvalue is zero to identify the member nodes for each cluster. The two eigenvectors for the corresponding zero eigenvalues are the first two columns of eigenvector **X**. Then, the cluster members can be found by identifying same elements in the vectors [0.5774, 0.5774, 0.5774, 0, 0, 0] and [0, 0, 0, 0.5774, 0.5774, 0.5774]. The first eigenvector shows that node 1, 2, and3 3 are in same cluster and second one shows the other cluster members (i.e., 4, 5, 6). The cluster member finding algorithm is illustrated in Table 13.

Table 13. Algorithm to identify the cluster members

```
% Find_Cluster_Mems (A)

% N – Number of nodes in the network
% L(·) – Laplacian Matrix
% EigVal_i(·) – i-th eigenvalue
% EigVec_ij(·) – Eigenvector associated with i-th eigenvalue
% N_CL_i – Number of nodes in i cluster
% SCLn_i – Set of nodes that are in cluster i

% Inputs: A
% Outputs: N_CL_i , SCLn_i

begin
   for All eigenvector associating eigenvalue is zero
      obtain first non-zero eigenvector value whose associating node is not visited
      set it to c
      N_CL_i = 0;
      for j = 1 to N
         if EigVec_ij (L) = c
            N_CL_i = N_CL_i + 1;
            SCLn_i(N_CL_i) = j;
         if node j is not visited
            c = EigVec_ij(L);
         end if
      end for
   end for
end
```

For example in Figure 38, consider the combined critical links (i.e., {B, F, G, J}). Once those nodes are removed from the network, the network partitioned into 3 clusters: $SCLn_1$ = {A, C, D, E}, $SCLn_2$ = {H}, $SCLn_3$ = {I, K, L, M, N, O} were $N_{CL\_1} = 4$, $N_{CL\_2} = 1$, $N_{CL\_3} = 6$. And all elements in cluster adjacent matrix are zeros such as $CL\_A_{ij} = 0$ for all $i, j = 1, 2, \ldots, N_{CL}$. The next step is to find the minimum cost possible links between clusters. Those possible additional links can be found from the cluster information and cost matrix in Table 12. In between cluster 1 and 2, the minimum cost of the possible additional link is 11 (i.e., between node E and H, $min\_C_{12} = 11$). Similarly, $min\_C_{13} = 12.1$ (i.e., between A and N) and $min\_C_{23} = 8.5$ (i.e.,

between I and H). Those obtained information can form the minimum cost matrix of possible links between clusters (i.e., *min_LC*)

5.2.2.2 CBMS algorithm

We propose 3 types of Cluster Based Merging Scheme in this section. The main idea of these three schemes is same but different conditions and node removal. The main idea is to determine the minimum number of additional links that reconnect the partitioned network due to removal of testing critical point. For each classified critical points, solve the problem such as

$$\textit{Solve: } \mathbf{min}\left(\sum_{i=1}^{N_{CL}}\sum_{j=1}^{N_{CL}}(PC\_A \times \min\ \_LC)\right) \wedge Eig_2\left(L(PC\_A)\right) \neq 0$$

Conditions:

$$\text{c1: } \sum_{i=1}^{N_{CL}}\sum_{j=1}^{N_{CL}}(PC\_A) = 2(N_{CL}-1)$$

where

PC_A –Adjacent matrix by testing combination of possible links between clusters

min_LC – minimum cost matrix of possible additional links between clusters

N_{lcl} – Number of local partitioned clusters

$L(\cdot)$ – Laplacina matrix

$Eig_2(\cdot)$ – Second smallest eigenvalue

The main algorithm of CBMS is illustrated in Table 14.

The first Cluster Based Merging Scheme (CBMS-1) finds the minimum cost possible links between clusters base on all nodes in the network while CBMS-2 excludes all critical points in the selection. For example in Figure 38, consider the critical point of node C. CBMS firstly remove the node C and the network partitioned into 2 clusters (i.e., $SCLn_1$ = {A, C, F, G,

H, I, J, K, L, M, N, O}, $SCLn_2$ = {D, E}). In the selection process of the minimum cost possible links between clusters, CBMS-1 considers all nodes except critical node B as a possible nodes for the additional links while CBMS-2 considers only non-critical nodes such as {A, D, E, G, I, O, N}. Then, CBMS-1 finds the cost minimal additional link between E and G whose cost is 6 by the cost matrix in table 12. However, CBMS-2 selects link between A and D whose cost is 8. Therefore, CBMS-2 may select more costly link but it reduces the number of possible nodes set and may reduce the number of additional link since it has more chance to share the additional link required to protect critical points. The last scheme, CBMS-3, finds the additional link with the removal of all critical points such as {B, C, F, G, J, K, L, M}. CMBS-3 may require more number of additional links than CBMS-1 or CBMS-2 but it is still survivable if multiple critical points fail simultaneously.

Table 14. Main Algorithm of Cluster Based Merging Scheme

% ***Import*** **Variable** from ***Find_Cluster_Mems*** **(*A*)**

% ***PC_A***– Adjacent matrix by testing combination of possible links between clusters
% ***min_LC*** – Minimum cost matrix of possible additional links between clusters
% N_{CL} – Number of clusters
% CL_A_i – Minimum cost additional links to protect critical point *i*

% **Inputs:** *A*
% **Outputs:** CL_A_i

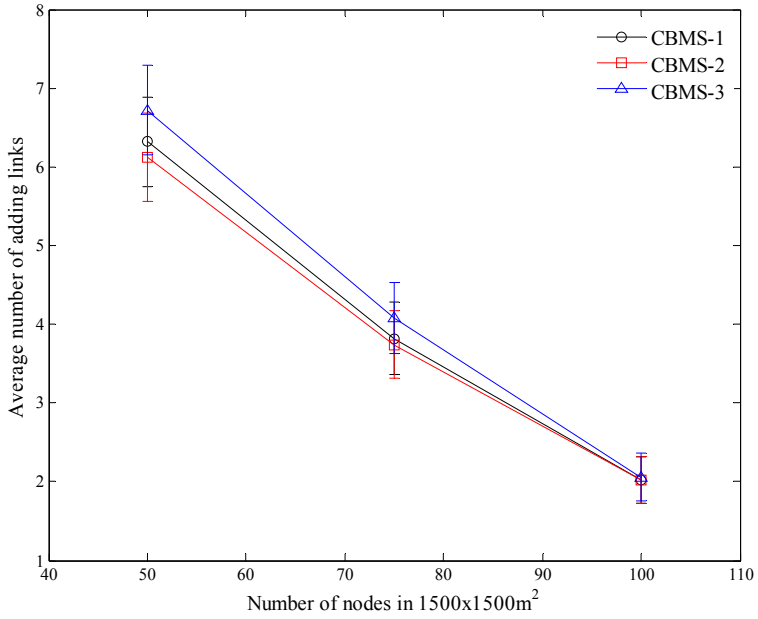
begin
  for All Classified Critical Points *i*
    remove set of nodes in testing critical point
    obtain result of *Find_Cluster_Mems* (*A*) and *min_LC*
    for all PC_A that satisfy $Eig_2(L(PC\_A)) \neq 0$ and $\sum_{i=1}^{N_{CL}} \sum_{j=1}^{N_{CL}} (PC\_A) = 2(N_{CL} - 1)$
      if $\sum_{i=1}^{N_{CL}} \sum_{j=1}^{N_{CL}} (PC\_A \times \min\ \_LC)$ is minimum
        $CL\_A_i = PC\_A_i$;
      end if
    end for
  end for
end

5.2.3 Numerical Study

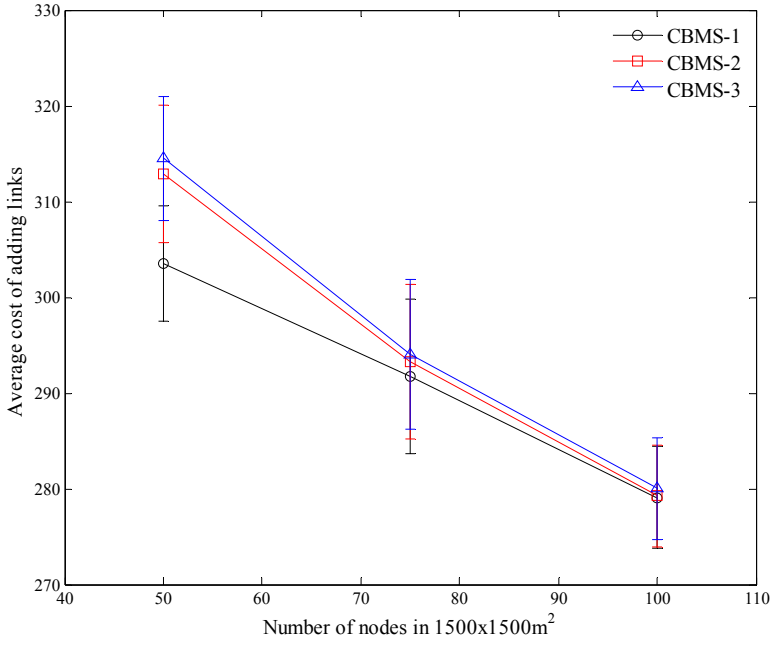
Here, we evaluate the effectiveness of our critical point management schemes (i.e., CBMS-1, CBMS-2, and CMBS-3) using simulation. We generate random topologies with different number of nodes (i.e., 50, 75, and 100) in a network area of $1500 \times 1500\ m^2$. The nodes are independently distributed according to a uniform [0-1500] random variable in the network area. For each network density, we generate 50 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. 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. All nodes are also assumed to have a capability to locate the node positions using GPS or localization techniques [36, 37]. In section, we compare those proposing 3 Custer Based Merging Schemes with average number of additional links and average cost to create those additional links. And also compare them with average node degree and average hop counts of the paths including the network without protection scheme. Figure 40(a) and (b) illustrates the average number of the additional links and average additional link cost. Average node degree and average hop count of the paths comparisons are in Figure 41(a) and (b).

According to Figure 40(a), the average numbers of additional links that are formed by those 3 CBMSs are comparable. The average number of additional links occurred by CBMS-1 is 6.32, by CBMS-2 is 6.12, and by CBMS-3 is 6.72 in 50 nodes network. Those average numbers of additional links are getting closer in denser network (i.e., 2.02 by CBMS-1, 2.02 by CBMS-2, 2.06 by CBMS-3 at 100 nodes network). However, CBMS-2 induces the least number of additional links while CBMS-3 induces the most in all network densities. CBMS-2 excludes the critical point in the selection of additional links and it may have more possibility for the critical

point to share the induced additional links with other critical points. CBMS-3 removes all the critical points and finds the cost minimal additional links to protect them. This process may require more additional links because it considers the multiple critical points failure case. The average cost of additional links in Figure 40(b) shows that CBMS-2 requires the least average additional link cost than the others where the cost is a distance between pair of end nodes of the additional link. It is more significant in sparse network (i.e., 303.52*m* by CBMS-1, 312.88*m* by CBMS-2, 314.5*m* by CBMS-3 at 50 nodes network). CBMS-1 tries to connect the network including other critical points at the absence of testing critical point while CBMS-2 excludes them. This may increase the additional link cost in CBMS-2. CBMS-3 shares the most of cost optimal links with CBMS-1 and 2 and it requires more additional links that may be more costly.

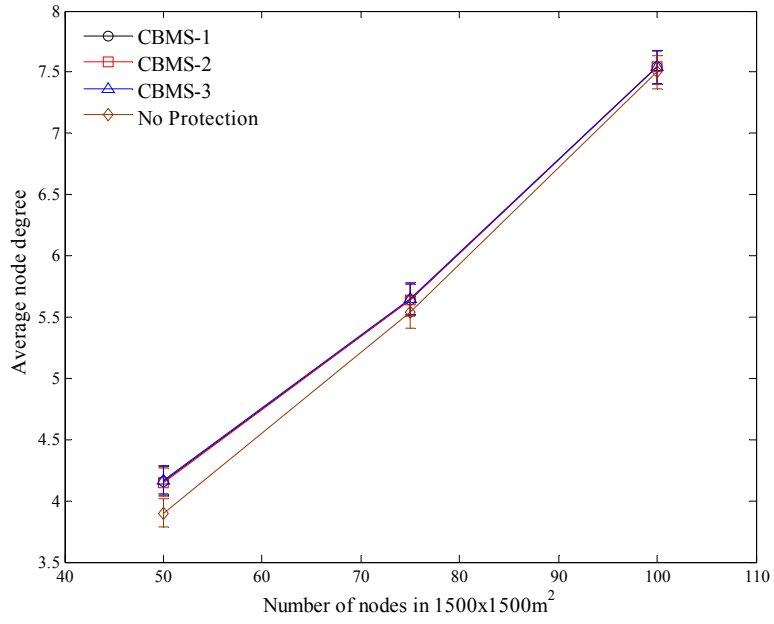


(a) Average Number of Additional Links

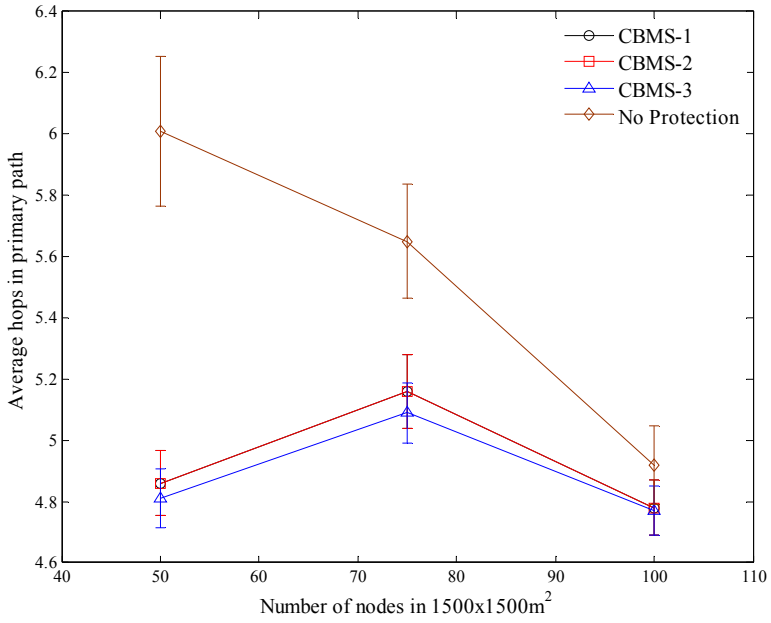


(b) Average Additional Link Cost

Figure 40. Comparison of Cluster Based Merging Schemes in average number of additional links and their average cost



(a) Average Node Degree



(b) Average Hop Count of the Paths

Figure 41. Comparison of Cluster Based Merging Schemes in average Node Degree and Hop Count of the Path including the network without the protection schemes

Average node degree increased by all 3 CBMSs is not significant. In Figure 41(a), the average node degree increment is 0.26 by CBMS-1, 0.25 by CBMS-2, and 0.27 by CMBS-3 at 50 nodes network. In 75 and 100 nodes network, those average node degree increments are even minimal (i.e., 0.1 or less for all schemes). Therefore, the link addition by all 3 schemes does not increase the overall node degree much. However, average hop count of the paths decreases significantly when those schemes are applied on critical points in sparse network. In 50 nodes network, the average hops of the paths is 4.86 by CBMS-1 and 2 and 4.81 by CMBS-3 while the initial average hops of the paths is 6.02. The difference decreases as the network is denser. In overall, CBMS-3 decreases the average hops of the paths the most. This is because one or more critical points disconnect the more than half of connectivity of pair of nodes. Here, we define the Disconnected Rate (DCR) as the ratio of disconnected pairs of nodes to all pair of nodes as

shown in equation (15), which indicates what percentage of total traffics are disconnecting when testing critical point fails.

$$DCR = \frac{\sum_{i=1,j=i+1}^{N_{CL}} N_{CL\_i} \times N_{CL\_j}}{n(n-1)/2} \quad (15)$$

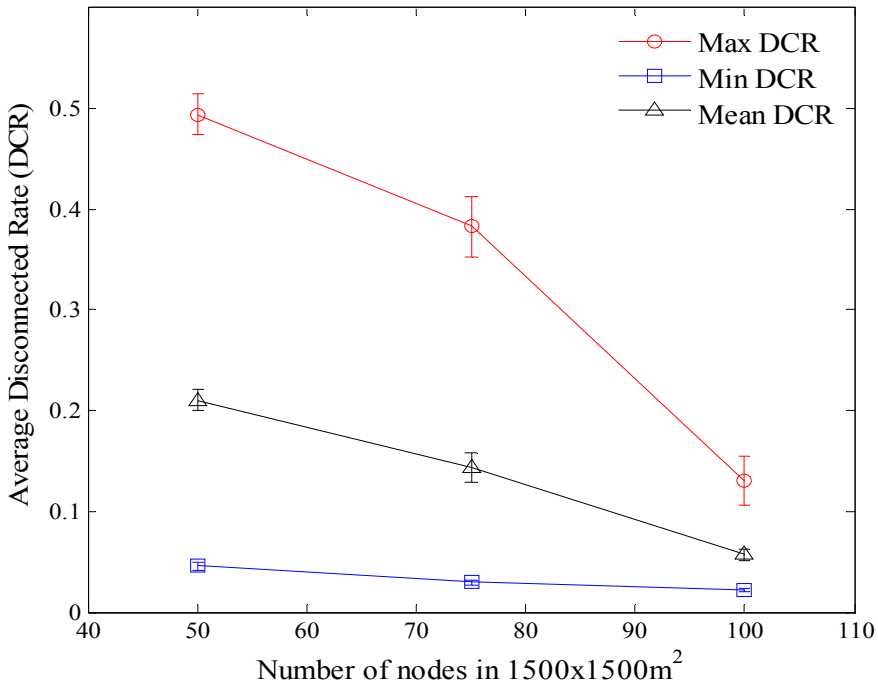


Figure 42. Max, min, and mean Disconnected Rate (DCR) over different network densities

A Figures 42 illustrates average of the maximum, minimum, and mean DCRs in different network densities. In 50 nodes network, the average maximum DCR reaches up to 0.5 and average mean DCR is 0.21. In 75 nodes network average maximum DCR drops to 0.38 and mean DCR drops to 0.14. DCR values are greatly dropped in denser network (i.e., 100 nodes network). These results explain why the average hop counts of the paths decreases significantly once those schemes are applied in sparse network. If the network partitions closely balanced due

to critical point failure, that critical point is bottleneck of the network. Therefore, the additional links that reconnects the testing critical point removed network provide the additional route for certain percentage of the traffics and average hop count of the paths reduces greatly.

Table 15. Average computation time comparison

N	CBMS-1	CBMS-2	CBMS-3
50	*0.0382s*	*0.0378s*	*4718.48s*
75	*0.0261s*	*0.0253s*	*1.3580s*
100	*0.1438s*	*0.1573s*	*0.2720s*

The average computation times of all 3 schemes are measured in Table 15. The average computation times of CBMS-1 and CBMS-2 are comparable while CBMS-3 takes significantly long time to find the cost optimized additional links. The computation time is measured for the time to solve the optimization problem described above. It takes more time when the network is partitioned into more number of clusters. Those numbers of clusters for each critical point removal in CBMS-1 and 2 are same but different links set to find additional links. However, CBMS-3 removes all critical points and it partitions the network into more number of clusters than CBMS-1 or 2. Then, CBMS-3 takes a lot longer time in sparse network and it decreases as the network is denser. In denser network, the network tends to be partitioned into small number of clusters when all critical points are removed and computation time reduces significantly.

5.3 DISCUSSIONS

We propose localized and globalized resilient schemes that protect the critical points and make the network 2-connected. In localized resilient scheme, the LFM scheme makes the network more survivable 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, LNLLC 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(2). However if the computation time is involved, LNLLC may not be eligible for the dynamic topology. If the network topology changes faster than the computation time of LNLLC, LFM will be the better solution. In this paper, we only consider no topology change or slow enough to ignore. When the local information is used for the critical point identification and protected using localized resilient schemes, the smaller H value induces higher average node degree and average additional link cost. This means that the more global topology information (i.e., higher H value) makes the localized resilient schemes protect the network with less average node degree and average additional link cost. However, the smaller local network information (i.e., smaller H value) protects the network at the most since it covers simultaneous failures of several multiple critical points while requires more cost and energy. However, localized resilient scheme may create unnecessary additional links since its topology information limitation. Some of critical points may share the same additional links. Those additional link sharing can be achieved by globalized resilient scheme (CBMS). Therefore, the globalized resilient scheme requires relatively less number of additional links while average cost may be expensive. In Cluster Based Merging Schemes, all CBMS-1, 2, and 3 shows similar results in average number and cost of additional links, node degree, and hop counts of the paths. CBMS-2 finds little less

average number of additional links while CBMS-1 is the most minimal average cost of additional links. In average hop count of the paths, CBMS-3 remarks the lowest. However, CBMS-1and 2 needs relatively shorter computation time and also CBMS-2 is still reliable when those single critical points are not available at the same time. Therefore, CBMS-2 may be the best choice in globalized resilient schemes.

5.4 CONCLUSIONS

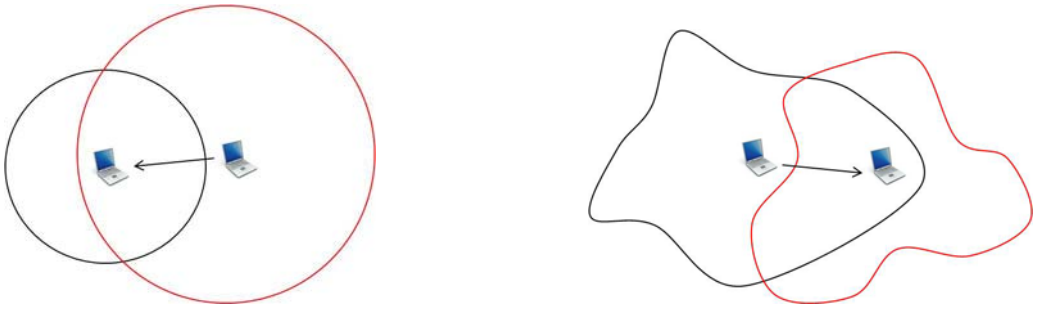
In this chapter, we propose localized resilient schemes with global network information and local network information. We also propose globalized resilient schemes. The resilient schemes on the critical points identified by local topology information increase the reliability of the network for the node and link failures while it requires more cost. Also localized resilient scheme creates more additional links than globalized resilient scheme. However, localized resilient scheme does not require knowing all network information while globalized resilient scheme does. Therefore, it is hard to decide which scheme is better. The resilient scheme selection should be selective to the application and what information of the network is available. For example, if the network requires less number of additional links and global network information is available, then the CBMS will be the best choice. However, if global network information is not available, then LNLLC should be selected. If the energy is not the mater but the reliability is the more concern, then LFM may be the better selection.

6.0 IMPROVING THE CONNECTIVITY OF HETEROGENEOUS MULTI-HOP WIRELESS NETWORKS

So far in this paper, we consider the homogeneous wireless. In homogeneous wireless network, the direct link between pair of nodes is symmetric, which means if one node can receive the packet from the other, so can the other. Many literatures study the network with a homogeneous network condition assumption. However, real wireless network is heterogeneous due to non-identical node condition, non-uniform transmission range, and etc. Although one node can receive the signal from the other, it does not mean that the other node can also receive the signal from this node (i.e., asymmetric link). For example, if one node is experiencing low battery power, then its transmission range is shorter than the one who has higher power and it make asymmetric link. In addition, the transmission range of the wireless node is not perfectly uniform and this also creates asymmetric links. In this chapter, we investigate the connectivity concerns in heterogeneous multi-hop wireless network. Firstly, we introduce the heterogeneous wireless network and its connectivity, which is different from the homogeneous. Secondly, we investigate the connectivity change of the network when its condition changes from homogeneous to heterogeneous. Then, we propose the cost effective schemes to reconnect the network that is partitioned due to the heterogeneous characteristic. Those schemes can effectively reconnect the network when they are suffering the connectivity problem from the heterogeneous network condition.

6.1 HETEROGENEOUS NETWORK

Many researchers have studied multi-hop wireless network connectivity determining conditions under which connectivity [1, 51] and k-connectivity [30-32] can be inferred probabilistically or assured asymptotically. 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. A major weakness of this work is the assumption of a homogeneous network context where nodes have identical properties and inhabit a uniform environment (e.g., identical transmission power, battery life, radio propagation ranges, antennas, etc.). Measurement studies [52] have shown that many of the assumptions in the homogeneous context are inaccurate. In particular it was noted that real networks can have directional links.



(a) Different Tx power levels (b) Non-uniform Tx range

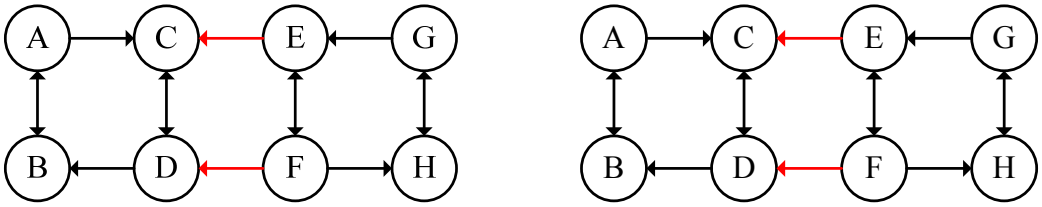
Figure 43. Directional links in heterogeneous wireless networks

For example, if each node has different transmission power, then their transmission range differs and a directional link can result as shown in Figure 43(a). Similarly, in Figure 43(b) a directional link results from nodes having non-identical signal propagation due to differences in the local environment (e.g., trees, buildings, etc.). In this section, we define the connectivity of

the heterogeneous multi-hop wireless networks and propose an algorithm to checks its network connectivity.

6.1.1 Heterogeneous Network Connectivity

A distinguishing characteristic of homogeneous networks is that the set of multi-hop paths between a pair of nodes are the same in each direction, while this may not be the case in a heterogeneous network.



(a) 1-Connected (b) Disconnected

Figure 44. Sample 8 node heterogeneous network topologies

For example, in the network of Figure 44(a), the path from node A to H is A→C→D→F→H whereas the path from H to A is H→G→E→C→D→B→A (i.e., $PATH_{A\rightarrow H} \neq PATH_{H\rightarrow A}$). Another effect of directed links in the topology is that communications may be one way. For example in the network of Figure 44(b), nodes A, B, C, and D can receive data from E, F, G, and H through the links E→ C and F → D. However, the opposite direction is not available such that A, B, C, or D, can communicate with E, F, G, or H. Given that directional links can occur in heterogeneous networks we define the connectivity between a pair of nodes as requiring that they be bi-communicable. Specifically we have the following.

Definition 1. Bi-communicable: *A pair of nodes is bi-communicable if and only if both nodes can receive information from each other. Note that bi-communicable doesn't*

require that paths between the nodes in question have the same set of intermediate nodes.

This definition indicates that node *i* and *j* are connected if and only if node *i* can receive data sent by *j* and vice versa. Based on this connectivity definition, we introduce a modified definition of a link between two nodes in a network.

Definition 2. Valid Link: *A pair of nodes i and j have a valid link if and only if a direct link exists in both directions or at least one direct link in either direction exists and at least one multi-hop path in the other direction is available.*

Based on Definitions 1 and 2 we define a partition of the network as follows.

Definition 3. Partitioned Network: *A network is considered partitioned if one or more nodes do not have bi-directional connectivity to the rest of the nodes in the network.*

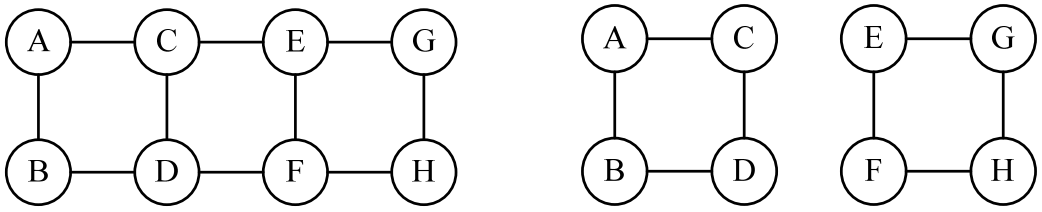
Hence a network is 1-*connected* if and only if all pairs of nodes have at least one path between them in *both directions*. In next section, we propose an algorithm for a network node to test the overall connectivity of the topology when directed links exist in the network.

6.1.2 Pre-Test of Network Connectivity

In this section, we propose the algorithm to test the connectivity of the network topology that includes the directed links. For the connectivity test, we use the same method of Algorithm I proposed in Chapter 4. Algorithm I uses the Algebraic connectivity, which is the second smallest eigenvalue of the Laplacian matrix defined in equation (9). If the Laplacian matrix of the topology has a single zero eigenvalue (i.e., if second smallest eigenvalue is positive → $N_{CL} = 1$). This connectivity test using algebraic connectivity is initially designed for the network whose

adjacent matrix is symmetric only. However, the heterogeneous wireless network adjacent matrix is not symmetric anymore, but asymmetric due to directed links. Thus, we need to transform the obtained asymmetric adjacent matrix into symmetric one to apply the network with the connectivity test.

The possible existence of directed links in the topology means that one can not directly apply the result. In order to force the adjacency matrix $A(t)$ to have a symmetric form we apply Definition 2 above in determining the link connectivity. Specifically, we require all links to be *valid* links and modify the link connectivity in a logically adjusted topology adjacency matrix $\acute{A}$ to reflect this. For example, the sample topologies of Figure 44 after adjustment will have the corresponding logical topologies shown in Figure 45. Note, that after adjustment of the network topology the resulting $\acute{A}(t)$ is symmetric and one can apply a test on the Laplacian eigenvalues to determine the network connectivity.



(a) Adjusted 1-Connected (b) Adjusted Disconnected

Figure 45. Adjusted network topologies corresponding to Figure 44

6.1.3 Heterogeneous Connectivity Test Algorithm

After the adjacent matrix transformation, it can be examine its connectivity using algebraic connectivity. The connectivity test procedure is given in algorithmic form in Table 16.

Table 16. Pseudo code of Heterogeneous Connectivity Test Algorithm (*h-CTA*)

Step 1	Identify any directed link in the topology from the adjacent matrix $A(t)$. If none, set $\acute{A}(t) = A(t)$ and go to step 3.
Step 2	Form adjusted adjacency matrix $\acute{A}(t)$ containing only valid links. Specifically, for each link $i - j$ such that $a_{ij} \neq a_{ji}$, $a_{ij} \in A$ $$if \begin{cases} a_{ij} = 1 \wedge Path(j \rightarrow i)\ exists, & a_{ij}' = a_{ji}' = 1 \\ a_{ji} = 1 \wedge Path(i \rightarrow j)\ exists, & a_{ij}' = a_{ji}' = 1 \\ \text{Otherwise}, & a_{ij}' = a_{ji}' = 0 \end{cases}$$ where $\acute{A}$ is the adjusted adjacent matrix.
Step 3	Compute the eigenvalues of the Laplacian matrix $L = \acute{D} - \acute{A}$, where $\acute{D}$ is the degree matrix corresponding to $\acute{A}$
Step 4	Determine the number of zero eigenvalues among the Laplacian spectrum. If $N_{CL} = 1$ then the network is connected, otherwise it is partitioned into N_{CL} components networks.

For a network of N nodes and E links of which K are directed, it can be shown that the time complexity of *h-CTA* is $O(K(E+N \log N)+N^2)$. The algorithm can be implemented at any network node having adjacency matrix information However, if the network is partitioned according to Definition 3, all nodes cannot exchange connectivity information. For example, in the network of Figure 44(b) cluster 2 (i.e., CL_2 = {E, F, G, H}) cannot receive connectivity information of cluster 1 (i.e., CL_1 = {A, B, C, D}) while cluster 1 can receive and obtain the connectivity information of cluster 2. Since the nodes in cluster 1 can obtain the global adjacency information, any node in cluster 1 can execute *h-CTA* to determine the overall connectivity.

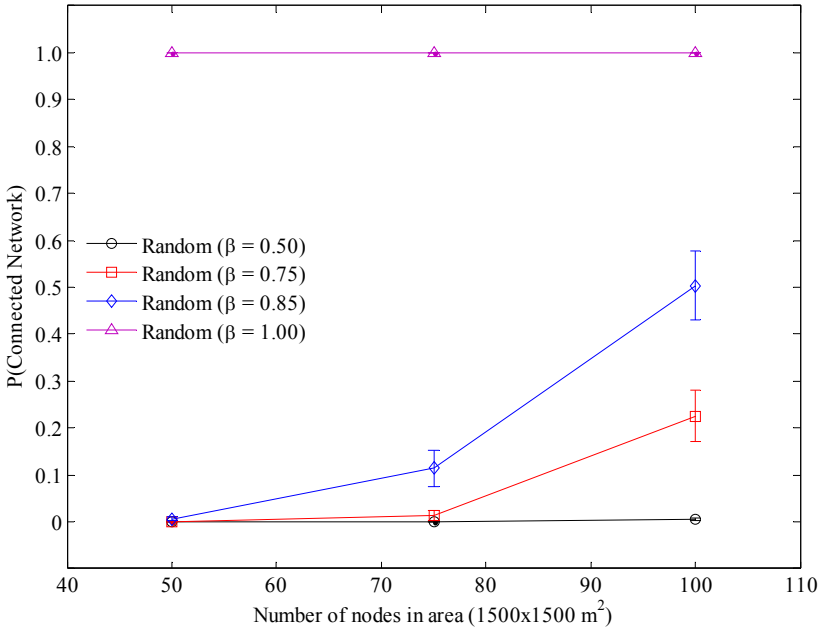
6.1.4 Numerical Study

We utilized the *h-CTA* algorithm to study the connectivity of heterogeneous multi-hop wireless network topologies. Here we discuss typical results for the case of random topologies. Using the ns2 simulator, we generated random topologies with different number of nodes (i.e., *50*, *75*, and *100*) in a network area of $1500 \times 1500\ m^2$. The nodes were independently distributed according

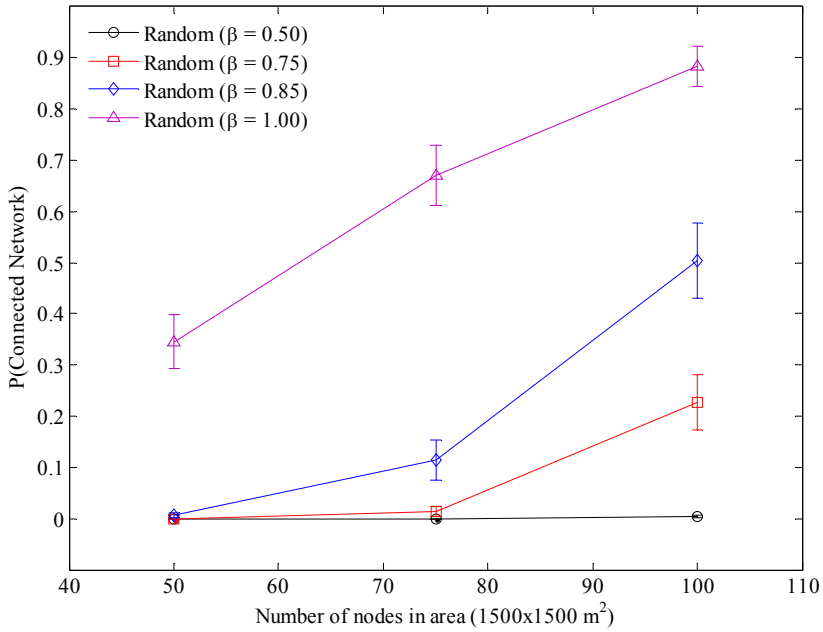
to a uniform [0-1500] random variable in the network area. We adopted baseline parameters from 802.11b equipment (i.e., average transmit power P_T = 15dBm, receiver sensitivity threshold P_{RSST} = -90 dBm). The basic transmission range was determined using a simple path loss model

$$P_R(\text{dBm}) = P_T(\text{dBm}) - 10\alpha\log(d) \tag{16}$$

where α is the path loss exponent and d is the distance between a pair of nodes in meters. A path loss exponent of α = 4.3788 was used, which results in a circular coverage area with radius 250 meters.



(a) Random networks with homogeneous transmission power



(b) Random networks with heterogeneous transmission power

Figure 46. Probability of connectivity in random network topologies

A heterogeneous network was created by varying either the transmission power or propagation model for each node or both factors. Each node *i* selects its transmission power P_{Ti} according to a uniform [13.5dBm - 16.5dBm] random variable. The resulting maximum transmission range R is uniform between 231m and 270.5m. Non-uniform signal propagation between nodes was modeled using the Quasi-Unit Disk Graph (Q-UDG) model. In the Quasi-UDG model, a link exists between two nodes if the inter-nodal distance d is less than βR, where R is the maximum transmission range of the node and β is the Q-UDG factor ($0 \leq \beta \leq 1$). For distances d greater than R, there is no connectivity. However, for $\beta R \leq d \leq R$, the link will exist with probability $(R - d)/(R-\beta R)$. We selected different β parameter values of 0.5, 0.75, 0.85 and 1.0 to show how irregular propagation affects the connectivity. Figure 46 shows 95% confidence intervals on the $P(\textit{connected network})$ determined with h-CTA versus the node density for two

cases of node power assignment: (a) homogenous with P_{Ti} = 15dBm at each node i and (b) heterogeneous with P_{Ti} drawn from a uniform [13.5dBm - 16.5dBm] random variable for each node. Each point in Figure 46 is determined from 4000 independent simulation runs. From Figure 46(a), notice that for the case of a homogeneous network (i.e., fixed P_T = 15dBm, β = 1.0) the network is connected (i.e., P(*connected network*) = 1) for all node densities considered. In contrast, the *P(connected network)* is almost zero for all network densities for both homogeneous (45(a)) and heterogeneous power assignments (45(b)) when β = 0.5. This is because the average transmission range is only 187.5 meters. However, the network connectivity increases as β increases. For example, in Figure 46(a) the probability of connectivity for a 100 node network is estimated from the simulation as 0.0043with β = 0.5, 0.226 with β = 0.75, and 0.504 with β = 0.85. Similarly for a fixed β the network connectivity increases with the node density, for example in Figure 46(a) with β = 0.85, the probability of connectivity is 0.0063 at 50 nodes, 0.1145 at 75 nodes, and 0.504 at 100 nodes. In comparing homogeneous power assignment with heterogeneous assignment (i.e., 45(a) vs. 45(b)) we see that only for the β = 1 case does the power assignment result in a significant difference in the connectivity. For the other values of β, the Q-UDG propagation effect dominates and the power assignment has little effect on the results (i.e., the confidence intervals on the results overlap). From Figure 46 one can clearly see that the connectivity in heterogeneous network is considerably lower than the homogenous case no matter what the cause of the heterogeneity.

6.2 CONNECTIVITY IMPROVEMENT SCHEMES

We propose schemes that for an unconnected heterogeneous network identify the links whose addition will connect the network. The approach is based on identifying the isolated groups (i.e. clusters) of nodes and determining the links required to connect between clusters. Consider a network partitioned into N_{CL} weakly connected clusters (i.e., each cluster has at least a directed link to some other cluster). Similar to Cluster Merging Scheme, the eigenvalues and corresponding eigenvectors of the Laplacian matrix L are used to identify the clusters and their member nodes as shown in Chapter5. Once the sets of member nodes in each isolated cluster are identified, a cluster adjacency matrix, CL_A, can be determined which represents how the clusters are asymmetrically connected. The cluster adjacency matrix can be determined by:

$$CL\_a_{ij} = \begin{cases} 1, & \forall\, k \in CL_i \wedge \forall\, l \in CL_j,\ \exists\, A_{kl} = 1 \\ 0, & \text{otherwise} \end{cases} \tag{17}$$

where CL_i is a nodes set of each cluster i, and A is the network adjacency matrix. The cluster adjacency matrix provides information about which cluster needs to be connected to another cluster to provide network connectivity. We propose two schemes to identify the links needed to reconnect the network based on given cost constraints: (1) Simple Merging Scheme (SMS) and (2) Cost Optimized Merging Scheme (COMS). The link cost is the cost to improve the link so that bi-communication can occur. Note that a variety of techniques such as node movement, transmission power, directional antenna, and etcetera can be used to add a link to the network. In this paper, we manipulate the transmission power to improve the links in question and the link cost constraint LC_{limit} in terms of distance is determined from the maximum transmission power together with the propagation model.

Table 17. Algorithm of Simple Merging Scheme (*SMS*)

```
% h-CTA – Heterogeneous Connectivity Test Algorithm
% N_CL – Number of clusters
% C_i – Cluster i where i = 1, 2, …, N_CL
% CLM_i – Member nodes of Cluster i,    i = 1, 2, 3, … N_CL
% N_CL – Number of clusters
% CL_A – Cluster adjacent matrix
% LC_ij – Cost of the possible links from cluster i to j
% MLC_ij – Minimum cost link from cluster i to j
% LC_limit – Maximum transmission range

% Inputs: A
% Outputs: addM    % Improvement required  links for the reconnection of the network

begin
   if  the network is partitioned using h-CTA
      obtain  N_CL, CL_A, CLM_i
      for  All i and j
         MLC_ij = Min{LC(k,l); k ∈ C_i, l ∈ C_j}
      end for

      for  All i and j  whose CL_A_ij = 0
         if  MLC_ij  <  LC_limit
            addM( i , j ) = 1;
         end if
      end for
   end if
end
```

6.2.1 Simple Merging Scheme (*SMS*)

The Simple Merging Scheme (SMS) tries to add links until each isolated cluster has a bi-directional connection with its neighbor clusters. The algorithm finds the set of *local* minimum cost links between an isolated cluster and a neighbor cluster using distance information. Specifically between each pair of neighbor clusters SMS finds the minimum distance links that are within the maximum possible transmission range as determined by LC_{limit} . The distance between nodes in isolated clusters can be obtained by Global Positioning System (GPS) or localization techniques [53, 54]. SMS can be implemented locally by the cluster heads (nodes

with a directed link to another cluster) in each cluster. SMS can be put in algorithm form as below in Table 17.

Table 18. Algorithm of Cluster Based Merging Scheme

```
% Import Variable from SMS (A)

% PL – Set of possible links that can be improved
% SL– Set of combination of selected links from PL
% TC– Total cost to improve the required links for reconnection

% Inputs: A
% Outputs: addM    % Improvement required links for the reconnection of the network

begin
   if the network is partitioned using h-CTA
      obtain N_CL, CL_A, CLM_i
      obtain PL = all 1s from CL_A – A
      set minTc = inf;
      for All combinations selected of links (SL) from PL
         if N_CL ≤ Number of SL ≤ 2( N_CL – 1) and Eig_2(L(CL_A + SL)) ≠ 0
            if TC( SL ) < minTC
               set all zero for addM
               for all i and j in SL
                  addM( i , j ) = 1;
               end for
               set minTC = TC( SL)
            end if
         end if
      end for
   end if
end
```

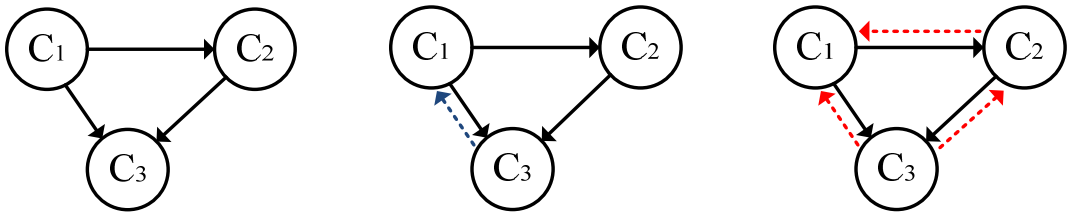
6.2.2 Cost Optimized Merging Scheme (*COMS*)

In order to provide a benchmark comparison to SMS we developed the Cost Optimized Merging Scheme (COMS) which finds the set of *globally* minimum cost links between isolated clusters, $MLC = \{\text{for all } i \text{ and } j;\ MLC_{ij}\}$ while satisfying the cost constraint LC_{limit}. This process is very time consuming since it has to check every combination of $2^{N_{CL}} - 1$ links and its computation

time is exponential. Therefore, before solving the optimization problem we add a step to merge small clusters with one node to the nearest larger cluster. COMS can be implemented in a centralized fashion at a super node and put in algorithm form as follows in Table 18.

6.2.3 Comparison of Cluster Merging Schemes

Figure 47 shows a simple example illustrating the differences between SMS and COMS. Figure 47(a) shows the original connections between the three clusters of a network. The nodes in cluster C_3 can obtain topology information from C_1 and C_2 and one can initiate COMS. Figure 47(b) shows the topological results of running the COMS algorithm. Observe that only a single directed link is added to provide bi-communicable connectivity to all clusters. In contrast the topology after running SMS is shown in Figure 47(c). Obviously SMS adds more links then COMS and thus costs more but it has the advantages of being distributed and making the network more robust to failures. Note that one node in C_2 and one in C_3 will independently initiate the SMS algorithm.



(a) Original clusters (b) COMS (c) SMS

Figure 47. Added links by SMS (RED) and COMS (BLUE)

6.2.4 Numerical Study

As a more extensive, evaluation and comparison of SMS and COMS with MND, we conducted a set of simulation based experiments similar to those reported in Section 6.1.4. Specifically, we generated random topologies with different node densities (i.e., *50*, *75*, and *100*) in a network area of $1500 \times 1500\ m^2$. The nodes were independently distributed according to a uniform [0-1500] random variable in the area. We adopted baseline parameters from 802.11b equipment (e.g., P_{RSST} = -90 dBm). Heterogeneous conditions were created by having each node *i* selects its transmission power P_{Ti} according to a uniform [13.5dBm - 16.5dBm] random variable and created non-uniform signal propagation between nodes using the Quasi-Unit Disk Graph (Q-UDG) model with $\alpha = 4.3788$, $\beta = 0.75$ or $\beta = 0.85$. We randomly generated topologies in this fashion until 50 weakly connected topologies were found for each node density. For each weakly connected topology both SMS and COMS were implemented to improve the connectivity. We examined three maximum node transmission powers constraints namely: 20dBm, 25dBm, and unlimited. In the unlimited power case for COMS the power was increased to the minimum power required to provide the minimum cost links necessary for bi-communicable 1-connectivity. In the unlimited power case for SMS, the power was increased until the minimum power necessary to add links that result in a full mesh of links between all clusters. While the unlimited transmit power case is impractical it provides a benchmark scenario for the results. For the evaluation and comparison, we use the MND technique. MND algorithm increases the transmission power to meet the minimum node degree requirement. Due to asymmetric links, we determine the minimum node degree by the minimum number of links in incoming and outgoing (i.e., MND = Min(din_{min}, $dout_{min}$)). For example, MND(2) represents that there are at least more than or equal to 2 of both incoming and outgoing links. For MND, we do not limit the

transmission power level (i.e., Tx = unlimited) in order for every node to meet the satisfaction of the minimum node degree requirement in both incoming and outgoing links. Firstly, we apply MND(d_{min}) technique to the topologies in order to obtain the d_{min} value which makes 100% connectivity for all topologies at all network densities (i.e., 50, 100, 150 nodes) at β = 0.75 and 0.85. Table 19 shows the probability of the connectivity when MND techniques are applied with different d_{min} values at both β values. It shows that the network becomes 1-connected for all nodes densities at d_{min} = 5 at both β. Since all topologies become 1-connected at d_{min} = 5, we choose it to compare with our schemes for both β values.

Table 19. Probability of 1-connectivity by MND(d_{min})

N	MND (d_{min} = 1)		MND (d_{min} = 2)		MND (d_{min} = 3)		MND (d_{min} = 4)		MND (d_{min} = 5)	
	β = 0.75	β = 0.75	β = 0.75	β = 0.75	β = 0.75	β = 0.75	β = 0.75	β = 0.75	β = 0.75	β = 0.75
50	0.00	0.02	0.40	0.48	0.86	0.90	1.00	1.00	1.00	1.00
100	0.00	0.08	0.54	0.62	0.90	0.94	1.00	1.00	1.00	1.00
150	0.00	0.26	0.58	0.60	0.88	0.88	0.94	0.98	1.00	1.00

Table 20. Average number of additional directed links by MND(d_{min})

N	MND (d_{min} = 1)		MND (d_{min} = 2)		MND (d_{min} = 3)		MND (d_{min} = 4)		MND (d_{min} = 5)	
	β = 0.75	β = 0.75	β = 0.75	β = 0.75	β = 0.75	β = 0.75	β = 0.75	β = 0.75	β = 0.75	β = 0.75
50	1.67	1.18	15.26	12.68	47.18	40.88	94.50	84.90	149.16	137.16
100	1.167	0.90	11.96	9.22	38.74	31.18	84.48	69.62	146.82	125.86
150	0.63	0.38	6.04	3.96	22.88	16.44	56.86	43.80	108.66	87.06

The corresponding number of additional directed links generating by the MND are shown in Table 20. MND generates many additional directed links as the d_{min} value increases for all nodes densities. At d_{min} = 5 which makes all 1-connected network, the average number of additional directed links is significantly large (i.e., 149.16 for β = 0.75 and 137.16 for β = 0.85 at 50 nodes). Comparing to our schemes, Figure 48 shows typical simulation results. For the three maximum transmit power limit investigates after applying SMS or COMS all topologies were at least bi-communicable 1-connected. Figure 48(a) and (b) show the average number of directed links added to the network versus network density for both SMS and COMS. As expected, when the network density increases fewer links are required to be added in order to provide connectivity regardless of whether MND, SMS, or COMS is used. For example, COMS adds an average of 5.52 links in 50 node networks and 1.96 links in 100 node networks with β = 0.75 and a maximum transmit power limit of 20dBm. Comparing MND(5) and our schemes, our schemes outperform in generating least number of additional directed links. For example, MND(5) generates averagely 149.16 of directed links by using unlimited transmission power to connect the network while SMS and COMS generate 7.38 and 5.52 of additional links with 20dBm of transmission limitation to connect the network at β = 0.75. Comparing SMS and COMS one can see that COMS needs fewer links to provide connectivity. In examining the effects of the maximum transmit power limit one can see that the two schemes behave differently. For SMS as the power increases more links are added beyond the minimum needed for 1-connectivity. In the extreme case of unlimited available transmit power, links are added until the clusters are interconnected with a full mesh of links. In contrast, COMS always chooses the cost optimized minimum number of additional links. Hence, the mean number of additional links decreases slightly with increasing maximum power limit. Further comparing the effect of β

on the results one can see that as β decreases and the signal propagation becomes more irregular the average number of directed links need to provide connectivity increases.

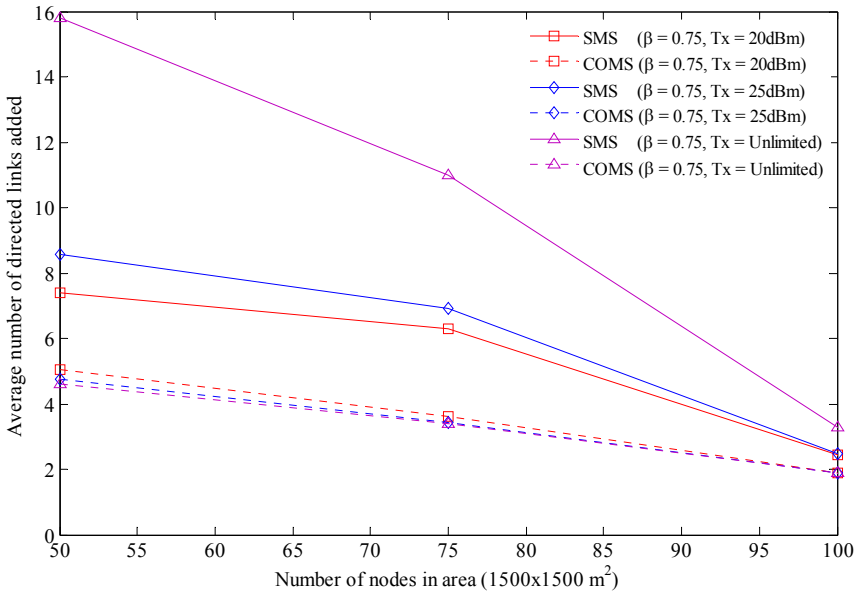
Figure 48(c) and (d) show how the average hop count in source-destination paths changes versus the network density for both schemes. The average hop count of all paths between every pair of nodes is considered. From the figures MND(5) provides significant shorter hop-count routes than our schemes does in all nodes densities for both β values (i.e., 3.67 by MND(5), 5.52 by SMS, 7.52 by COMS in 50 nodes network with β = 0.75 for unlimited Tx) because it creates considerably large number of costly links (i.e., larger transmission power) and they shorten the hop-count for every pairs of nodes. Comparing our schemes, SMS outperforms COMS for all network densities, β values, and restoration restrictions. For example, with a 20dBm maximum power level restriction, the average hop count for SMS is 6.72 and COMS is 7.35 for a 50 node network with β = 0.75. Since COMS finds the minimum number of additional links to provide connectivity, while SMS generate as many as it could, the average hop count by SMS should be always smaller than COMS. On the other hand, in terms of the average link cost COMS always results in a lower average cost and the difference is more significant as the maximum power level limit increases. Our simulation studies show that our schemes outperform MND technique to make heterogeneous network topology be 1-connected. Our schemes require significantly a lot less number of additional directed links.

The time complexity of MND is $O(N^2)$ since it checks every node's node degree in each direction where N is the number of nodes. Note that, the worse case maximum number of directed links to connect N_{CL} clusters is $2(N_{CL} - 1)$. The time complexity of SMS can be shown to be $O(N_{CL}^2)$ and that of COMS is $O(k \times N_{CL}^2)$ where k is a number of possible link

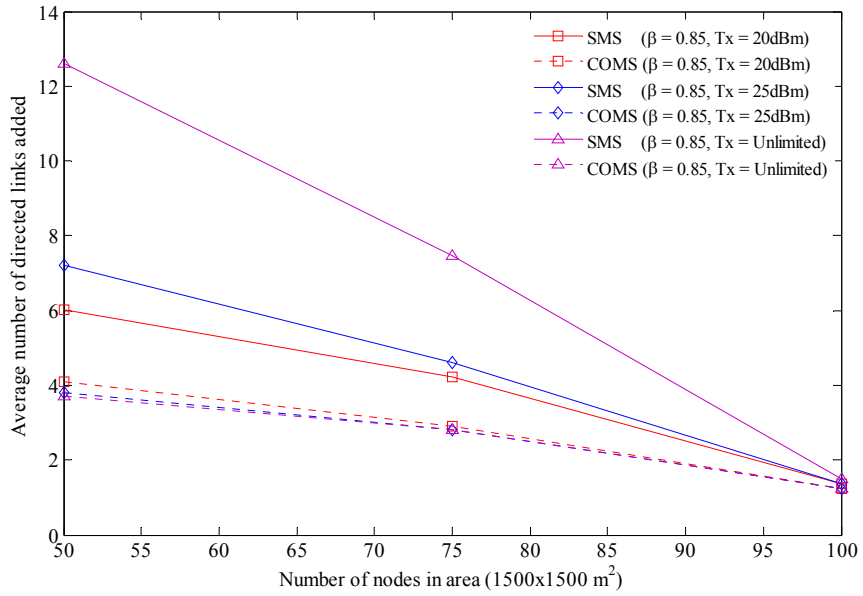
combinations. The range of the number of link combinations is 1 to 2(N_{CL} – 1) and k can be computed by

$$k = \sum_{l=1}^{2(N_{CL}-1)} \binom{lmx}{l}, \quad \text{where, } lmx = 2\binom{N_{CL}}{2} \tag{18}$$

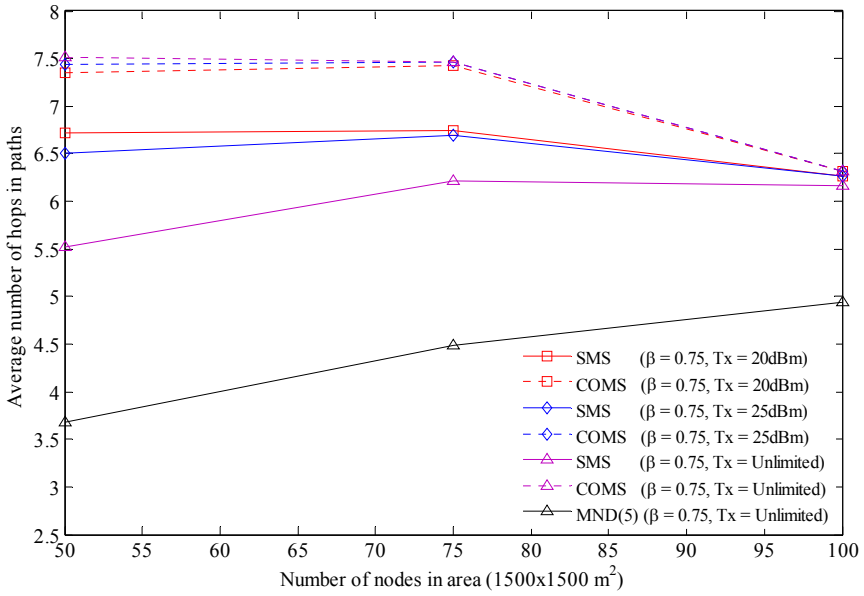
Therefore, the computation time of SMS is smaller than that of COMS. If the number of isolated clusters N_{CL} increases, the computation time of COMS increases more quickly than SMS.



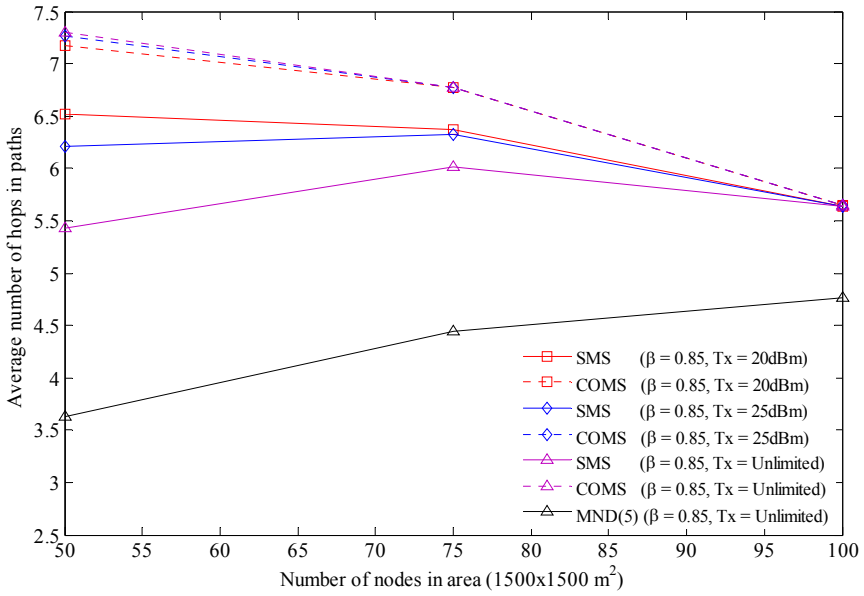
(a) Average number of directed links added for $\beta = 0.75$



(b) Average number of directed links added for $\beta = 0.85$



(c) Average number of hops in paths for $\beta = 0.75$



(d) Average number of hops in paths for $\beta = 0.85$

Figure 48. Comparison of SMS and COMS for average number directed links and Comparison of MND, SMS, and COMS for average hops in paths

6.3 IMPLEMENTATION OF RESILIENCE SCHEMES AND PERFORMANCE

We propose the resilient schemes that improve the connectivity in both homogeneous and heterogeneous wireless networks. In homogeneous wireless network, free space radio propagation model, of which the radio signal attenuates by the distance, is used. Thus, the connectivity of the pair of nodes is determined by the distance. Unlike homogeneous, the connectivity in heterogeneous wireless network depends on not only distance between nodes but also the random environmental interference. As mentioned in this chapter, the directed link is generated due to heterogeneity. Quasi-UDG model is used for the network connectivity and the numerical study indicates that the heterogeneity induces the directed links and it easily

disconnects the network especially in sparse network. Generally, asymmetric links are not considered as valid link in routing protocol, which results in network performance degradation. In this section, we implement our proposed cluster merging scheme to the network and examine the network performance in non-uniform transmission range condition via simulation study.

6.3.1 Network Model

In homogeneous wireless, the radio propagation depends on the distance between pair of nodes as in equation (15) in 6.1.4. All nodes have an identical uniform unit disk transmission range. The link is establishing between any nodes in this range. The signal strength is attenuated only by the distance in this network condition. Normally, free space or 2 ray-ground propagation model is used. The connectivity between pair of nodes is determined by the distance. Thus, those links are bi-directional in general. However, in real wireless network environment, there are many obstacles, which induce the random attenuation of the signal. This random attenuation in the path loss model is called shadow fading, *X*, as shown in equation (19).

$$L_p = L_0 + 10\alpha\log_{10}(d) + X \tag{19}$$

where $L_0 = 10\log_{10}(P_t) - 10\log_{10}(P_0)$ (i.e., P_t is transmitting power, P_0 is the receiving power at 1 meter), d is a distance, and X is a random variable with distribution. The measurement study indicates that this random variable follows log-normal distribution where its probability density function is shown in equation (20).

$$f_{LN}(x) = \frac{1}{\sqrt{2\pi}\sigma x}\exp\left(\frac{-(\ln x-\mu)^2}{2\sigma^2}\right) \tag{20}$$

where μ is the mean received signal strength and σ is its standard deviation. Standard deviation σ is the factor that determines the randomness of the shadow fading effect.

6.3.1.1 Shadow Fading Effect

The shadow fading path loss effects heavily on the network connectivity. As illustrated in Figure 46(a) in chapter 6, the probability of the network being connected due to the random effect of environmental interference with homogeneous transmission power is zero for any non-zero value of β in sparse wireless network (i.e., 50 nodes network). We implement the shadow fading propagation model in the network and the connectivity between pair of nodes is probabilistic based on equation (19). Normally, α is between 2.7 and 3.5 [57] and σ can be up to 15 dB [58].

Figure 49 illustrated the probability of connectivity between two nodes in distance where the path loss, α = 2.7, standard deviation, σ = 4 dB, transmission power, P_t = 24.5 dBm, and the receiving signal strength threshold $Rx_{threshold}$ = - 64.37 dBm. The probability of the packet delivering dramatically decreases as the distance is larger than 100*m* approximately and almost zero around 325*m*. When the node distance is 150*m*, the packet delivery rate is less than 0.4 while it is 1 for free space or 2 ray-ground propagation model. This implies that the link quality is degraded with the probability by the node distance. In later part of this chapter, the shadow fading effects on the network performance is studied using simulation.

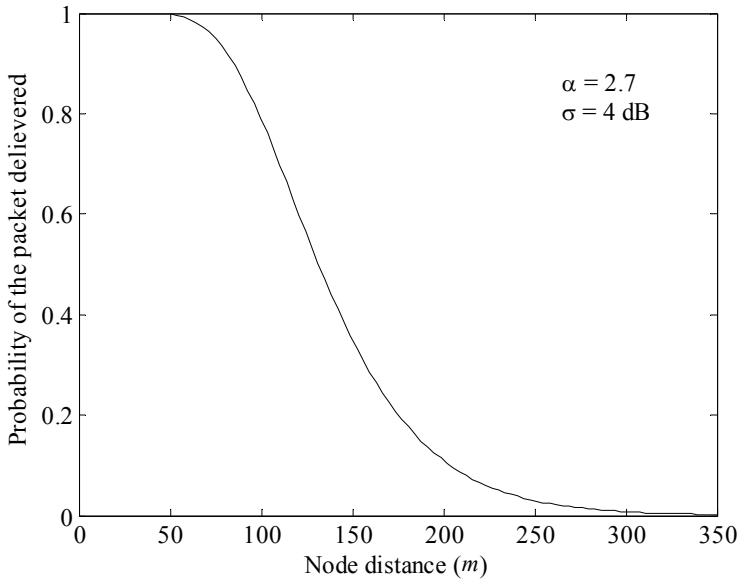


Figure 49. Packet delivering probability of shadow fading propagation model in distance where $\alpha = 2.7$, $\sigma = 4$ dB, $P_t = 24.5$ dBm, and $Rx_{threshold} = -64.37$ dBm

6.3.1.2 Routing protocol

In this study, we use the Optimized Link State Routing Protocol that utilizes the expected transmission count metric (ETX) [22]. This proactive protocol periodically updates the link quality and selects the best quality path. Each node broadcasts certain number of packets and counts the retransmission to compute the delivery ratio as in equation (21) where d_f is the forward delivery ratio and d_r is the reverse delivery ratio.

$$\mathrm{ETX} = \frac{1}{d_f \times d_r} \tag{21}$$

6.3.2 Simulation Study

In this section, we use NS-2 simulator to perform the simulation study. First, we examine how the heterogeneous network condition has an effect on the network performance using two

different propagation models (i.e., 2 ray-ground and Shadow fading), that represent homogenous and heterogeneous wireless network. Then, we evaluate our schemes using simulation.

6.3.2.1 Simulation setup

We randomly generate the 40 different 50 nodes topologies. The nodes are uniformly distributed over the area of $1500\times1500m^2$ and topologies are at least 1-connected in UDG model. The transmission power level of each node is set to 24.5dBm. The receiving threshold is set to -64.4dBm, which means that the receiver receivers the packet if the received signal strength at the receiving is greater than the threshold and this threshold makes the transmission rage of $250m$. We use 802.11 for the MAC layer protocol and OLSR or the Routing protocol. We randomly generate 1000 bytes of Constant Bit Rate (CBR) traffics and CBR is generating with the rate of 0.25. CBR packets are delivering via User Datagram Protocol (UDP). The number of traffic connection is randomly selected by use of the mean number of connections (MC), which is the ratio of the number of connections to the number of total possible connection (i.e., $n(n\text{-}1)/2$). The priority of the traffic connection is given to the pair of nodes that has a larger hop-count (i.e., longer path) in order to illustrate more effective performance evaluation focused on bad quality links. The starting time of CBR traffic is randomly selected before simulation time of 10 seconds where the total simulation time is 60 seconds. Then, the throughput is measured from 10 seconds to 60 seconds of simulation time. The throughput is computed by dividing the number of received packets by given time (i.e., 50 seconds of simulation time). In the simulation study, we set everything same with random number of random traffics to compare.

6.3.2.2 Comparison of propagation models

As shown in Figure 46, the network connectivity drops a lot at the heterogeneous network condition (i.e., Quasi-UDG with $0 < \beta < 1$) even with the homogeneous transmission power. In this section, we use shadow fading propagation model to illustrate the radio propagation in the heterogeneous network environment such as quasi-UDG in previous section while 2 ray-ground is used for the homogeneous. We use at least 1-connected topologies to run the simulation and compare the 2 ray-ground and shadow fading propagation models. We set the parameters of 2 ray-ground in order to set the transmission range of 250m approximately. Transmitting and receiving antenna gains (i.e., G_t, G_r) are set to 1.0 where the frequency is 914MHz. The height of transmitting and receiving antenna are set to 1.5 for both. For shadow fading model, we use pathloss exponent of $\alpha = 2.7$ and deviation of $\beta = 4$dB.

Figure 50 illustrates the average throughputs with 95% of confidence interval at each MCs (i.e., MC =0.1, 0.3, 0.5). Red line represents the average throughput of the network by 2 ray-ground and blue line represents the results by shadow fading propagation model. The results show that the network throughput drops significantly when the shadow fading is applied. For example, the throughput drops from 304.73 packets/s to 35.337 packets/s at MC = 0.1. Similar throughput drops are shown at MC = 0.3 and 0.5. The difference of the throughput is 269.39 packets/s at MC = 0.1, 313.14 packets/s at MC = 0.3, and 312.16 packets/s at MC = 0.5. These results indicate that the network is degraded when the shadow fading is applied.

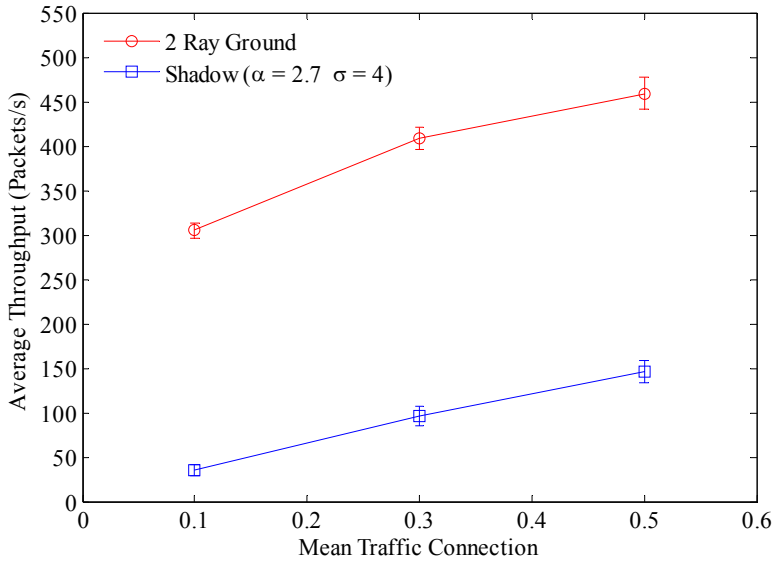


Figure 50. Average throughput in two different propagation model in MC = 0.1, 0.3, 0.5

6.3.2.3 Improving network performance

Previous section shows that the shadow fading degrade the network performance significantly. We then control the transmission power of selected nodes to improve the link quality in order to increase the delivery rate. First, we identify the set of critical points of network topology using transmission range of 250*m*. Second set of transmission power control nodes are identified using Cluster Based Merging Scheme based on adjacent matrix measured by ETX in OLSR. CBMS finds the nodes that will connect the network that is partitioned by the ETX. At each selected node, we increase the transmission power up to 33dBm.

Figure 51 illustrates how the average throughput is improved by transmission power control on selected nodes. It shows that CBMS increase the throughput the most at MC = 0.1 and 0.3 (i.e., 67.16 and 128.93 packets/s). At MC = 0.5, improvements by CBMS and critical points are similar (i.e., 158.61 by critical points and 158.96 by CBMS). This represents that the critical points degrade the network performance when its link quality is low; however there still exist

several set of links that degrades the network when their link quality is low. Figure 52 illustrates the average number of nodes that are controlled their transmission power in CBMS. It observed that the more nodes are modified at higher MC.

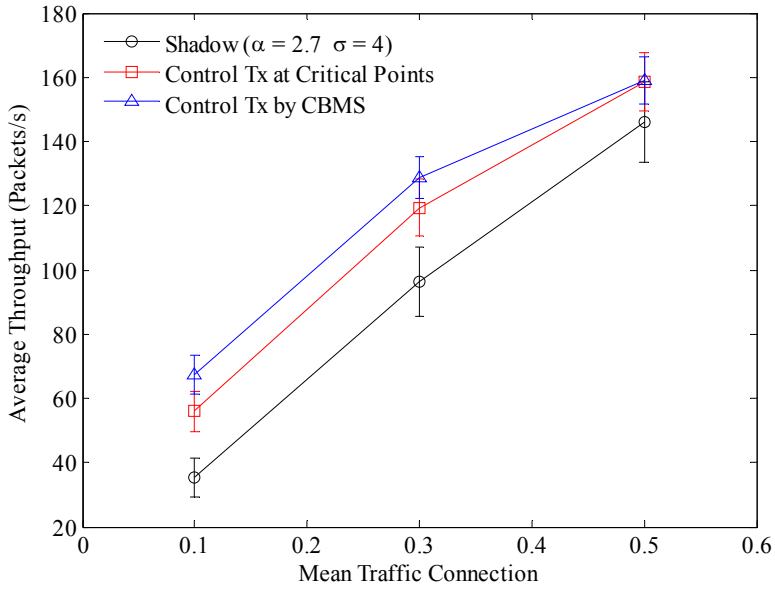


Figure 51. Average throughput improvements by Tx control at MC = 0.1, 0.3, 0.5

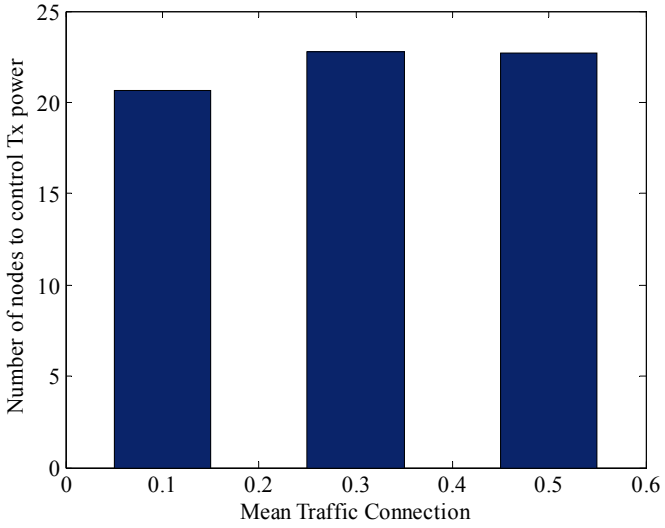


Figure 52. Average number of nodes to control Tx by CBMS at MC = 0.1, 0.3, 0.5

6.3.2.4 Homogeneous vs heterogeneous transmission power under shadow fading

In this section, we compare the throughput change in different network conditions in different transmission power and environment scenarios such as homogeneous Tx with 2 ray-ground (HoT2R), heterogeneous Tx with 2 ray-ground (HeT2R), homogeneous Tx with shadow fading (HoTSf), and heterogeneous Tx with shadow fading (HeTSf). We uniformly distributed 20 nodes network over $680\times680m^2$, which are at least 1-connected. All other simulation parameters are same as in previous setup. Figure 53 shows the differences of average throughputs in different network conditions. The result indicates that the shadow fading has a most effect on throughput degradation.

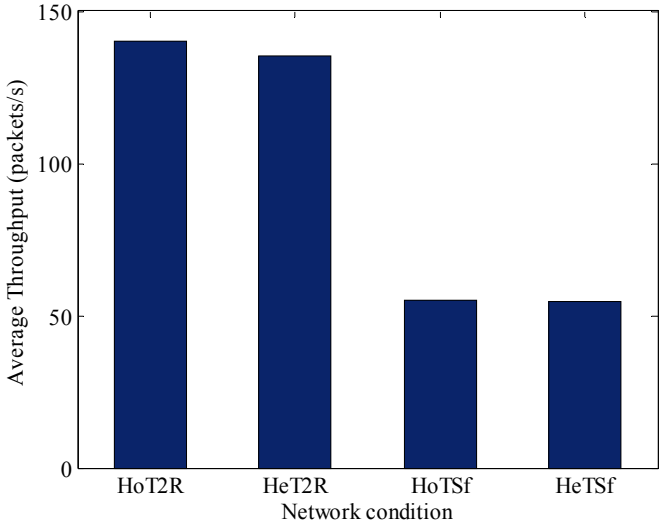
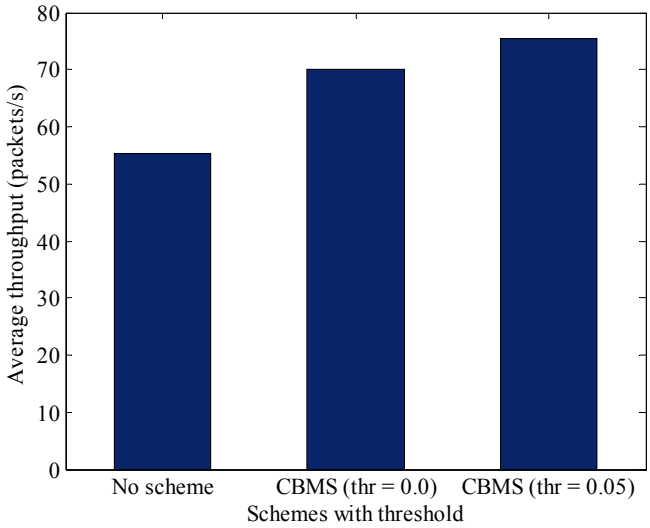


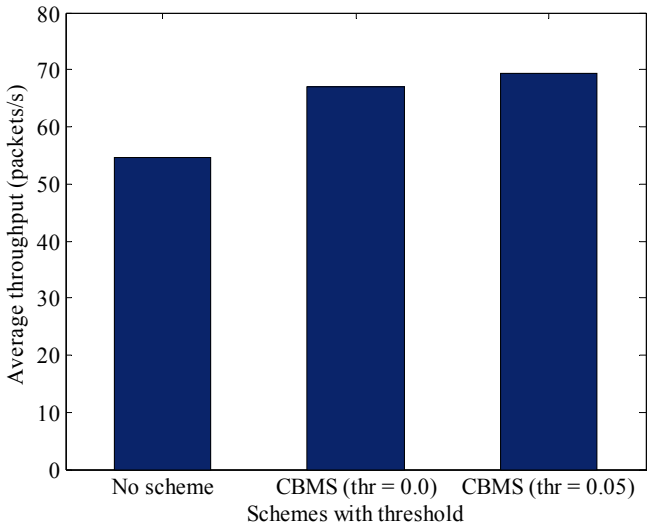
Figure 53. Average throughputs for different network conditions at homogeneous Tx with 2 ray-ground (HoT2R), heterogeneous Tx with 2 ray-ground (HeT2R), homogeneous Tx with shadow fading (HoTSf), and heterogeneous Tx with shadow fading (HeTSf)

Based on the degraded network, we use the CBMS to identify the nodes that may improve the performance by transmission control. We use threshold concept in this study. Firstly, we identify the nodes by CBMS using the ETX matrix measured by OLSR. In this case,

we find the links of which link quality is 0. The other approach is threshold. We set threshold and remove all links whose link quality is lower than threshold in the ETX matrix. Then, we use modified ETX matrix to find the nodes for Tx control.



(a) Homogeneous Tx and shadow fading condition



(b) Heterogeneous Tx and shadow fading condition

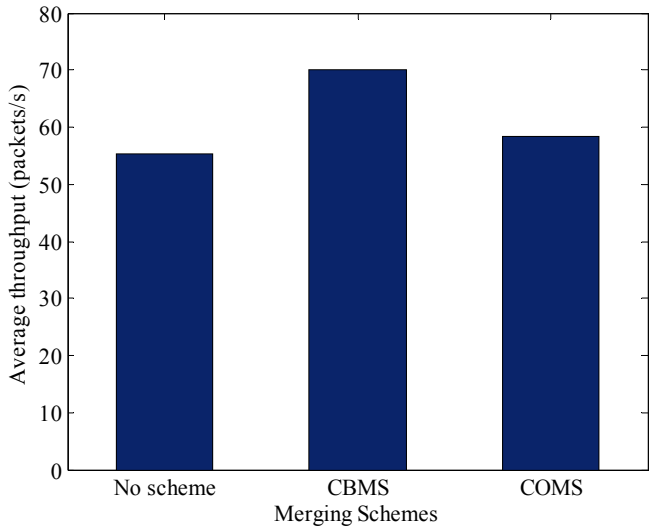
Figure 54. Average throughputs of no Tx controlled and Tx controlled by CBMS with threshold of 0 and 0.05 in homogeneous and heterogeneous Tx in shadow fading condition

Figure 54 illustrates the throughput improvements by CBMS with 0 and 0.05 thresholds to the ETX matrix for the network condition of homogeneous Tx and heterogeneous Tx in shadow fading. CBMS is performed based on the ETX matrix obtained by threshold restriction and the throughput improves gradually when the threshold increases (i.e., 70 packets/s by thr = 0 and 75 packets/s by thr = 0.05 in HoTSf). Therefore, the CBMS with higher threshold improves the network performance better. However, the higher threshold produces more number of clusters, which causes the higher computation time.

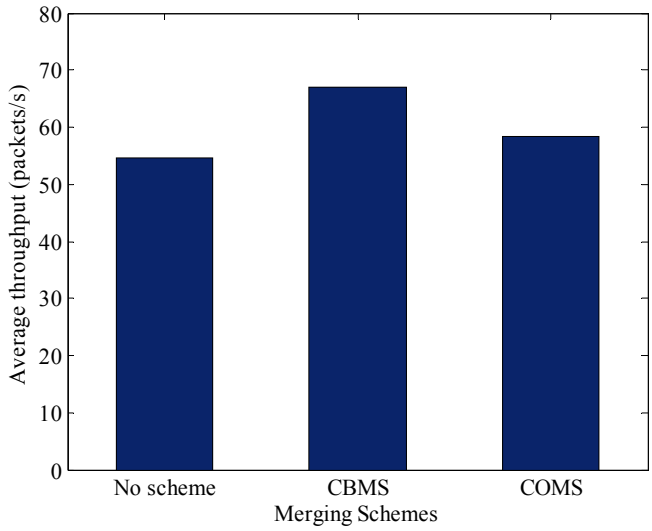
6.3.2.5 Asymmetric links

OLSR uses the link quality computed by the equation (20), which means it only uses the symmetric links. However, there still exist the asymmetric links that we can use it to reduce the number of nodes to control Tx. In this section, we use the asymmetric matrix measured at MAC layer. In OLSR, each node send HELLO packet periodically. Then, each node counts the number of received HELLO packet for certain period of time to compute the uni-directional link quality. By use of obtained asymmetric link quality matrix, we identify the node to control the Tx using Cost Optimized Merging Scheme (COMS). It finds out the uni-direction link to be improved and we control the source node to create the asymmetric link. Creation of asymmetric link makes the link symmetric and traffic can be delivered through this link by OLSR. Figure 55 illustrates the average throughput in homogeneous and heterogeneous Tx with shadow fading network condition and improved average throughputs by CBMS and COMS. The results show that the throughput improvement by COMS is small for both network conditions. Generally, if the links is asymmetric, it may have bad link quality because link quality is usually related to distance. Since COMS improves only uni-directional link, the link in the other direction remains still bad even if it is shown valid in measured link matrix. Therefore, it does not guarantee the sufficient

link improvement compare to CBMS. However, COMS reduces the number of Tx controlled nodes averagely by 1.75 nodes in HoTSf and 1.725 nodes in HeTSf.



(a) Homogeneous Tx in shadow fading condition



(b) Heterogeneous Tx in shadow fading condition

Figure 55. Average throughputs comparison between CBMS and COMS in network condition of homogeneous and heterogeneous Tx in shadow fading condition

6.4 DISCUSSIONS

The definitions in this chapter are useful for the heterogeneous multi-hop wireless network. Also the Heterogeneous Connectivity Test Algorithm (*h-CTA*) based on those definitions is very effective the connectivity test for those networks. It checks the bi-communication of the directed links only. This algorithm is much faster than the traditional connectivity examining algorithm, which checks both directions of all possible pairs of nodes. From the numerical study, it is found that at least 1-connected network is more likely disconnected when the heterogeneous network condition is applied. Utilizing the connectivity testing algorithm (*h-CTA*) to obtain transformed adjacent matrix and the eigvenvalues and eigenvectors of its Laplacian matrix, two merging schemes are proposed. Simple Merging Scheme (SMS) makes the cluster matrix bi-connected while Cost Optimized Merging Scheme finds the minimum cost links to reconnect the network. We compare our schemes with MND technique to connect the network. According to numerical study, our schemes make the network at least 1-connected by selecting significantly small number of additional directed links. For the comparison between our schemes, COMS does increase the cost of the improving links for both variation of β and LC_{limit}. However, its computation time is relatively high. Meanwhile, SMS increase the cost of links as the β and LC_{limit} vary and its increment is larger at sparser network. But its computation time is relatively small. Therefore, the selection of the connecting scheme should be selective. If the network topology changes relatively fast, SMS should be used while COMS is useful if the topology does not change often.

We also evaluate the network performance by the throughput in heterogeneous network using simulation study. The simulation results show that the improvement is not significant. This

is because the network size is small and the traffic load is not heavy. If the network size is bigger and traffic load is heavier, the improvement may increase.

T he simulation results indicate that the network performance degrades dramatically in heterogeneous network. Our CBMS scheme improves the average throughputs by transmission power control. It is observed that the heterogeneity of transmission power degrades the network more compare to the shadow fading. Ours CBMS with threshold increase the average throughputs, but it requires higher computation time since threshold produces more number of clusters. We also investigate how the network performance change when asymmetric link are considered by COMS. It reduces the number of Tx controlled node. But the throughput improvement is minimal. The asymmetric link becomes symmetric link by COMS, but it has high chance that the link quality of this asymmetric link may be low as well. This may cause minimal improvement by the COMS.

6.5 CONCLUSIONS

Many 1-connected homogeneous wireless networks become disconnected when the network becomes heterogeneous. In this paper we have proposed a new algorithm to determine the connectivity of heterogeneous wireless networks. The results of a simulation based numerical study utilizing our proposed algorithm to examine the effects of several factors (variations in power levels, irregular signal propagation, and network nodal density) on connectivity are presented. Further we propose two connectivity management schemes SMS and COMS to add additional links to a partitioned heterogeneous network in order to provide at least 1-connectivity. A simulation based study of the two schemes shows that both schemes can correctly add links to

provide connectivity, but SMS is easier to implement with the drawback of a higher cost. They also effectively select the least number of links to make the network at least 1-connected compare to the MND technique. We also evaluate the performance of the network. Our scheme relieves the network degradation due to network heterogeneity. However, COMS shows minimal improvement. Further study can be investigating more about the COMS to improve the performance. One solution may be the combined schemes of CBMS and COMS.

7.0 CONCLUSIONS AND FUTURE WORK

In this paper, we examine the connectivity assumption of MANETs used in several literatures. It is observed that the assumption, the probability that the minimum node degree is k is greater than or equal to the probability that the network is k-connected, is not valid especially in sparse MANETs. It is also observed that it is not easy to achieve and maintain the certain number of minimum node degree in sparse MANETs and the minimum node degree does not guarantee k-connected network. This is because there exist one or more of critical points; nodes, links, or both. Therefore, we approach this problem in different view point. In this study, we assume that the node mobility is none or minimal to ignore so that the network topology is stationary for enough time. Beyond our study, many extensive works are providing in this chapter.

7.1 CONTRIBUTIONS

This dissertation has examined the connectivity and resilience issues of the wireless ad hoc and sensor network and the major contributions are followings.

- We propose the weak point approach to the network topology of the MANETs where the weak point is the node, link, or combination that partitions the network for its failure. We, then, present the importance of the weak points in the network.

- We propose two heuristic algorithms to identify weak points. We implement the graph theory in Algorithm I and neighbor nodes connectivity in Algorithm II.
- We study the critical node's behaviors in different scenarios such as network density, mobility, critical node's positions, multiple critical nodes, and global critical node finding by *H*-hop subnetwork information.
- We propose the local resilience schemes such as Local Full Mesh (LFM) and Least Number of Links with Least Cost (LNLLC). They effectively reinforce the critical points in homogeneous wireless network condition.
- We propose the global resilience schemes Cluster Based Merging Schemes (CBMSs) that find the lesser number additional links than local resilience schemes utilizing global topology information in homogeneous network condition.
- We examine the connectivity of the heterogeneous wireless network, which contains the asymmetric links. Our numerical study presents that the connectivity degrades significantly in the heterogeneous wireless network condition.
- We propose two connectivity recovery schemes, Simple Merging Scheme (SMS) and Cost Optimized Merging Scheme (COMS). They find the direct links that can recover the at least 1-connectivity in heterogeneous wireless network condition.
- We evaluate CBMS and COMS to improve the performance of the network utilizing Optimize Link State Routing (OLSR) protocol. Our schemes improve the throughput of the network.

7.2 FUTURE WORK

In this study, several future works can be interesting. Firstly, the time complexity of Algorithm I is relatively high. One possible future work can be reducing its computation time by reducing the set of testing points.

The cost metric in the local resilience schemes is the distance between nodes in this study. However, the computation time for each scheme is different. It can be combining in cost metric to select the optimal resilience scheme for the network condition.

In heterogeneous wireless network connectivity study, we assume that the network is at least 1-connected in homogeneous wireless network as an initial condition. If the network is not at least 1-connected in homogeneous wireless network condition, our schemes are not valid for connectivity recovery. Therefore, it has to identify the additional links to connect the network that is initially partitioned in homogeneous wireless network condition. Several selected nodes in each cluster increases its transmission power to send its cluster information to other possibly existing nearby clusters.

The performance improvement study using COMS will be more significant if other link quality improvement methods are used such as directional antenna. COMS does not improve the performance significantly due to the existence of the existing poor links. COMS with threshold method can improve the performance.

APPENDIX

NS-2 SIMULATOR VALIDATION

Network Simulator version 2 (i.e., NS-2) is open source simulator, which is a discrete event simulator for the network [59]. It simulates the network events such as creating packets, routing, sending, receiving, forwarding, and etc. in timely manner. Many researchers use this simulator to study the network behaviors upon certain conditions for research purpose. NS-2 simulator is used in this dissertation to examine the network behaviors such as impact of critical node failure in Section 3.3.1 and performance measure in Section 6.3.2. In order to support the results for these studies, here, we provide the NS-2 simulator validation.

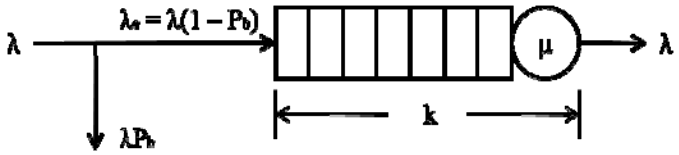


Figure 56. Queue model of M/M/1/k

A simple queue model such as M/M/1/k is used for the validation. The assumptions for M/M/1/k queue model are followings. The packet arrival follows Poisson process with average rate of λ and the mean service time μ is exponentially distributed. The service is First In First Out

(FIFO) fashion and only one server exists with limited capacity of k. The typical queue diagram is shown in Figure 56.

Due to the limited size of capacity, M/M/1/k queue drops the packets arrived when the capacity is full. The packet loss rate, Pb shown in Figure 56, can be computed by equation 21 where ρ is server utilization.

$$P_b = \frac{(1-\rho)\rho^k}{1-\rho^{k+1}}, \qquad \text{where } \rho = \frac{\lambda}{\mu} < 1 \tag{21}$$

In the simulation, we assume that the packets arrive according to a Poisson process with mean rate λ = 30 packets/sec where the exponentially distributed mean service time μ = 33 packets/sec. The link speed is 100 kbps. Then, we randomly generate the packet size with average of 378.78 bytes. Since the server utilization is less than 1 (i.e., ρ = 0.909), we use equation (21) to compute the packet loss of the system. For the packet loss comparison, we run NS-2 simulation for 1000 seconds of simulation time and measure the number of sent and dropped packets to compute packet loss by dividing the number of dropped packets by sent packets. We compare packet loss rate in different k values (i.e., k = {2, 3, 4, 5, 6, 7, 8, 9, 10}). In each k value, we perform 100 times of simulations and average packet loss rates are shown in Figure 57 with 95% confidence intervals.

The black solid line is the packet loss rate compute by the equation and red line is from the simulation results. It shows that the simulation results by NS-2 are very close to the packet loss rate by the equation for all k values. Therefore, the NS-2 is valid to use it for simulation study.

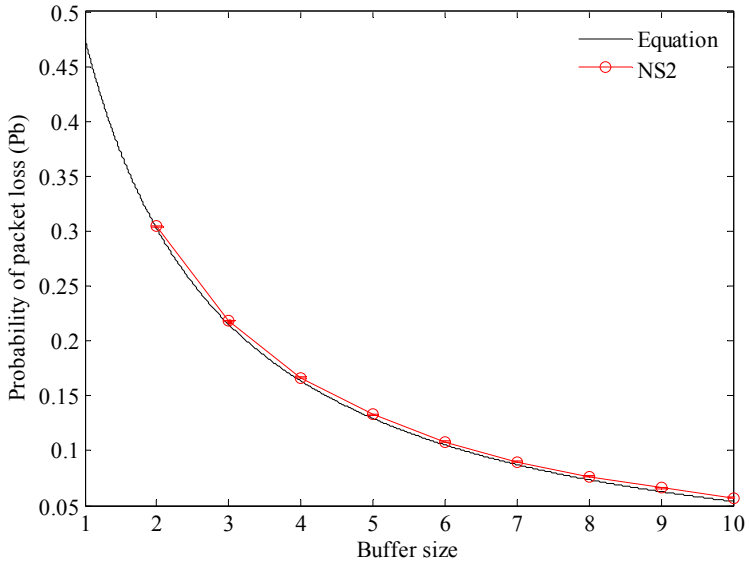


Figure 57. Comparison of probability of packet loss by analytical model and simulation results

BIBLIOGRAPHY

[1] I. Chlamtac, M. Conti, and J. J.-N. Liu, "Mobile ad hoc networking: imperatives and challenges", *Elsevier B. V.*, 2003.

[2] J. Koroma, W. Li, and D. Kazakos, "A Generalized Model for Networks Sruvivability," *TAPIA'03,* October 15-18, 2003.

[3] D. Y. Chen, S. Garg, And K. S. Trivedi, "Network Survivability Performance Evaluation: A Quantitative Approach with Applications in wireless Ad-hoc Networks," *MSWiM'02,* pp.61-68, September 28, 2002.

[4] D. Tipper and T. Dahlberg, "PCS Network Survivability", *IEEE*, 1998.

[5] D. Tipper, T. Dahlberg, H. Shin, and C. Charnsripinyo, "Providing Fault Tolerance in Wireless Access Networks," *IEEE Commun. Mag.*, vol. 40, no. 1, Jan. 2002, pp. 58–64.

[6] Sterbenz, James P.G., Krishnan, Rajes, Hain, Regina Rosales, Jackson, Alden W., Levin, David, Ramanathan, Ram, and Zao, John, "Survivable Mobile Wireless Networks: Issues, Challenges, and Research Directions," *WiSe'*02, September 28, 2002.

[7] T. Dahlberg and K. Subramanian, "Visualization of Real-time Survivability Metrics for Mobile Networks", *ACM*, 2000.

[8] G. Di Caro and M. Dorigo, "AntNet: a mobile agents approach to adaptive routing", *Tech. Rep. IRIDIA/97-12, Universite' Lebre de Bruxelles,* Belgium.

[9] R. Beckers and J. L. Deneubourg, "Trails and u-turns in the selection of the shortest path by the ant lasius niger," *Journal of Theoretical Biology*, 159, 397-415.

[10] E. Bonabeau, M. Dorigo, and G. Theraulaz, "Swarm Intelligence," *From Natural to Artificial Systems*. New York: Oxford University Press, 1999.

[11] S. Rajagopalan and Chien-Chung Shen, "ANSI: A Unicast Routing Protocol for Mobile Ad hoc Networks Using Swarm Intelligence," *Proc. International Conference on Artificial Inteligence,* pp.24-27, 2005.

[12] P. Arabshahi A. Gray, I. Kassabalidis, A. Das, S. Narayanan, M. El-Sharkawi, and R. Marks II, "Adative Routing in Wireless Communications Networks using Swarm Intelligence," http://sensorweb.jpl.nasa.gov.

[13] A. Nasipuri and S.R. Das, "On-demand multi-path routing for mobile ad hoc networks," *Proc. IEEE ICCCN 1999,* pp. 64-70, 1999.

[14] S. J. Lee and M. Gerla, "Split Multipath Routing with Maxmally Disjoint Paths in Ad Hoc Networks," *Proc. IEEE ICC 2001,* pp. 3201-3205, 2001

[15] M. Pearlman, Z. Haas, P. Sholander, S. Tabrizi, "Alternate Path Routing in Mobile Ad Hoc Networks," *Proceedings of IEEE MILCOM 2002.*

[16] S. J. Lee and M. Gerla, "AODV-BR: Backup Routing in Ad hoc Networks," *IEEE,* 2000.

[17] A. B. McDonald and T. F. Znati, "A Path Availability Model for Wireless Ad-Hoc Networks," *Proceedings of IEEEWireless Communications and Networking Conference*, 1999.

[18] S. Jiang, D. He, and J. Rao, "A Prediction-based Link Availability Estimation for Mobile Ad Hoc Networks," *INFOCOM*, pp. 1745–52, 2001.

[19] R. Leung, J. Liu, E. Poon, A.-L. C. Chan, and B. Li. "MPDSR: A QoS-aware Multi-path Dynamic Source Routing Protocol for Wireless Ad-Hoc Networks," *IEEE LCN'01*, pp. 132–141, 2001.

[20] S. Guo, O. Yang, Y. Shu, "Improving Source Routing Reliability in Mobile Ad Hoc Networks," *IEEE Computer Society,* 2005.

[21] A. Trivino-Cabrera, I. Nieves-Perez, E. Casilari, F. J. Gonzalez-Canete, "Ad Hoc Routing Based on the Stability of Routes," *ACM MobiWAC'06*, October 2, 2006.

[22] D. S. J. Couto, D. Aguayo, J. Bicket, R. Morris, "A High-Throughput Path Metric for Multi-Hop Wireless Routing," *Wireless Networks*, 2005. **11**(4): p. 419-434.

[23] R. Draves, J. Padhye, and B. Zill, "Routing in Multi-Radio, Multi-Hop Wireless Mesh Networks, " *ACM New York*, 2004, NY, USA.

[24] P. Gupta and P.R. Kumar, "Critical Power for Asymptotic Connectivity in Wireless Networks," *Stochastic Analysis, Control, Optimization and Applications,* Boston: Birkhauser, pp. 547-566, 1998.

[25] P. Santi, D. M. Blough, and F. Vainstein, "A probabilistic analysis for the range assignment problem in ad hoc networks," *MobiHoc* 2001.

[26] P. Santi and D. Blough, "An evaluation of connectivity in mobile wireless ad hoc networks," *DSN* 2002.

[27] P. Santi and D.M. Blough, "The Critical Transmitting Range of Connectivity in Sparse Wireless Ad Hoc Networks," *IEEE Transactions on Mobile Computing,* Vol 2. No. 1, January-March 2003.

[28] F. Xue and P.R. Kummar, "The Number of Neighbors Needed for Connectivity of Wireless Networks," *Wireless Networks 10*, Kluwer Academic, Netherlands, 2004, pp. 169-181.

[29] R. Hekmat and P. V. Mieghem, "Study of connectivity in wireless ad-hoc networks with an improved radio model," *In Workshop on Wireless Optimization, WiOpt,* 2004.

[30] 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.

[31] Z H. Zhang and J. C. Hou, "On the critical total power for asymptotic k-connectivity in wireless networks," in *Proc. IEEE INFOCOM*, March 2005.

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

[33] X. Li, P. Wan, Y. Wang, and C. Yi, "Fault Tolerant Deployment and Topology Control in Wireless Networks," *ACM MobiHoc*, Annapolis, Maryland, 2003.

[34] M. D. Penrose, "The longest edge of the random minimal spanning tree," *ANNALS OF APPLIED PROBABILITY*, 1997. **7**: p. 340-361.

[35] M. D. Penrose, "On k-connectivity for a geometric random graph," *Random Structures & Algorithms*, 1999. **15**(2): p. 145-164.

[36] M. Jorgic, I. Stojmenovic, M. Hauspie, D. Simplot-Ryl, "Localized algorithms for detection of critical nodes and links for connectivity in ad hoc network," *Proc.. 3rd IFIP MedHoc,* Bodrum, Turkey, June 27-30, 2004, 360-371.

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

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

[39] B. Milic and M. Malek, "Adaptation of the Breadth First Search Algorithm for Cut-edge Detection in Wireless Multihop Networks," *MSWiM*, Chania, Crete Island, Greece, October, 2007.

[40] T. Clausen and P. Jacquet, "The optimized link state routing protocol," (RFC 3626), www.ietf.org/rfc/rfc3626.txt, October, 2003.

[41] C.D. Waal and M. Gerharz, "Bonnmotion: A mobility scenario generation and analysis tool," 2003. Available: http://www.cs.uni-bonn.de/IV/BonnMotion/

[42] T. Lewis, "Critical Infrastructure Protection in Homeland Security: Defending a Networked Nation, " *Wiley-Interscience*, 2006

[43] C. Godsil and G. Royle, "Algebraic Graph Theory," *Springer*, 2001.

[44] X. Hou and D. Tipper, "Impact of Failures on Routing in Mobile Ad Hoc Networks Using," *DSR*, 2003.

[45] P. Brutch and C. Ko, "Challenges in intrusion detection for wireless ad-hoc networks," *IEEE Workshop on Security and Assurance in Ad hoc Network*, 2003.

[46] T. Anusas-amornkul, "On Detection Mechanisms and Their Performance for Packet Dropping Attack in Ad Hoc Networks," 2008.

[47] P. Basu and J. Redi, "Movement control algorithms for realization of fault-tolerant ad hoc robot networks," *Network, IEEE*, 2004. **18**(4): p. 36-44.

[48] Z. Han, A. Swindlehurst, and K. Liu, "Smart deployment/movement of unmanned air vehicle to improve connectivity in MANET," *WCNC*, 2006.

[49] W. D. Grover, "Mesh-Based Survivable Transport Networks: Option and Strategies for Optical, MPLS, SONET and ATM Networking", *New York: Prentice-Hall*, 2003.

[50] C. Perkins, "Ad hoc on demand distance vector (AODV) routing," Internet Engineering Task Force-Draft, draft-ietf-manet-aodv-04.txt, Oct., 1999.

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

[52] D. Kotz, C. Newport, R.S. Gray, J. Liu, Y. Yuan, and C. Elliott, "Experimental Evaluation of Wireless Simulation Assumptions", *Proceedings of the 7th ACM International Workshop on Modeling Analysis and Simulation of Wireless and Mobile Systems (MSWiM 2004),* Venice, Italy, Oct., 2004.

[53] 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.

[54] 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.

[55] S. Das, H. Liu, A. Kamath, A. Nayak, and I. Stojmenovic, "Localized Movement Control for Fault Tolerance of Mobile Robot Networks," *International Federation for Information Processing* (*IFIP*), 2007.

[56] P. Basu and J. Redi, "Movement control algorithms for realization of fault-tolerant ad hoc robot networks," IEEE Network, 18(4):36-44, 2004.

[57] T.S. Rappaport, "Wireless Communications," *Principles and Practice*, Prentice Hall, 1996.

[58] N. Geng and W. Wiesbeck, "Panungsmethoden fur die Mobikommunikation," *Springer*, 1998.

[59] The Network Simulator-ns-2; http://www.isi.edu/nsnam/ns/