Design, Analysis and Resource Allocations in Networks

In Presence of Region-Based Faults

by

Sujogya Banerjee

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Approved January 2013 by the
Graduate Supervisory Committee:

Arunabha Sen, Chair
Guoliang Xue
Andrea Richa
Glenn Hurlbert

ARIZONA STATE UNIVERSITY

May 2013

ABSTRACT

Communication networks, both wired and wireless, are expected to have a certain level of fault-tolerance capability.These networks are also expected to ensure a graceful degradation in performance when some of the network components fail. Traditional studies on fault tolerance in communication networks, for the most part, make no assumptions regarding the location of node/link faults, *i.e.*, the faulty nodes and links may be close to each other or far from each other. However, in many real life scenarios, there exists a strong spatial correlation among the faulty nodes and links. Such failures are often encountered in disaster situations, *e.g.*, natural calamities or enemy attacks. In presence of such region-based faults, many of traditional network analysis and fault-tolerant metrics, that are valid under *non-spatially correlated* faults, are no longer applicable. To this effect, the main thrust of this research is design and analysis of robust networks in presence of such region-based faults. One important finding of this research is that if some prior knowledge is available on the maximum size of the region that might be affected due to a region-based fault, this piece of knowledge can be effectively utilized for resource efficient design of networks. It has been shown in this dissertation that in some scenarios, effective utilization of this knowledge may result in substantial saving is transmission power in wireless networks.

In this dissertation, the impact of region-based faults on the connectivity of wireless networks has been studied and a new metric, region-based connectivity, is proposed to measure the fault-tolerance capability of a network. In addition, novel metrics, such as the region-based component decomposition number(RBCDN) and region-based largest component size(RBLCS) have been proposed to capture the network state, when a region-based fault discon-

nects the network. Finally, this dissertation presents efficient resource allocation techniques that ensure tolerance against region-based faults, in distributed file storage networks and data center networks.

DEDICATION

To my Baba (Father) and Ma (Mother) and loving sisters for all their supports and inspirations.

ACKNOWLEDGEMENTS

I would like to offer my sincere regards and thanks to my advisor Dr. Arunabha Sen for his supervision, guidance and advices throughout my doctoral studies. I am thankful for his constant encouragement and support throughout the program. I would also like to extend my gratitude to Dr. Guoliang Xue, Dr. Andrea Richa and Dr. Glenn Hurlbert for serving on my committee.

I am immensely grateful to my seniors Dr. Sudheendra Murthy (Sudhiji) and Dr. Pavel Ghosh (Pavel da), whose precious advices and guidance made my initial years of doctoral studies smooth. Special thanks to my colleague and co-author Shahrzad. I had a fun and enjoyable experience working with her. I would like to thank my friend Ayan for his constructive feedback on my thesis. I would also like to thank my lab-mates Anisha, Zahra, Harsh, Arun and many others who made this experience enjoyable.

Finally, I would like to thank my parents and sisters whose constant encouragements and supports helped me to finish my doctoral thesis successfully. I would specially thank my elder sister (Bordibhai) for motivating me and inspiring, right from childhood, to reach this goal. At the end, I would like to thank all my friends who made my stay in Arizona enjoyable and memorable.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

INTRODUCTION

1.1 Introduction

In the last few decades the world has seen remarkable advancements in computer networks, both in wired and wireless domain. Today we find the presence of computer networks in the form of large fiber-optic cables running across continents as well as small sensor networks collecting and processing data of different environmental parameters. Developments in computer networks has lead to a more connected world than before. With this huge progress in computer networks, it has also become extremely critical to ensure robustness of the networks. Robustness of a network means a high level of fault tolerance capability in the networks against any kind of failures (node/link failures). Also in any unfortunate incident, even if number of faults exceeds this level, the networks are expected to ensure a *graceful degradation* in performance. Traditional studies on fault tolerance in wired and wireless networks, for the most part, do not assume any spatial correlation among the faulty nodes and links. In other words, no assumptions regarding the location of node/link faults are made, *i.e.*, the faulty nodes and links may be close to each other or far from each other. However, in most of the real-life scenarios, nodes and links failures do not appear randomly across the networks. Instead there exists a strong spatial correlation among the faulty nodes and links and they are most likely to be confined within a limited area or region. Some examples of such localized faults in wired and wireless networks are as follows:

- Due to natural calamities like earthquakes, tsunamis, forest fires, hurricanes or floods, communication nodes and associated links inside that particular disaster region may get destroyed. As a result telecommunication networks at the center of the disaster region may get disconnected creating a communications “black hole". Failures of such network devices are geographically correlated. For example, the U.S. Gulf Coast’s telecommunications infrastructure, heavily concentrated in the New Orleans metropolitan area, suffered extensive damage during the Category 5 storm Hurricane Katrina on 2005 [1].

- In a war zone an EMP attack or an enemy bomb can destroy communication infrastructure of the region.

- Sensor nodes located at a certain part of the wireless network may die due to energy depletion for heavy traffic [2]. Also by performing jamming attacks on the wireless networks an attacker may create a jamming hole in certain part of the network making the nodes in that part inoperable [3].

This situation is shown in Figure 1.1 where the shaded part indicates the fault region. In this thesis such localized faults are termed as *region-based faults*. Study of fault-tolerance in communication networks in presence of *region-based faults* is significantly different from the fault-tolerance study in presence of *random faults*.

The two immediate questions that arise in the context of the study of *region-based fault* tolerant metrics are:

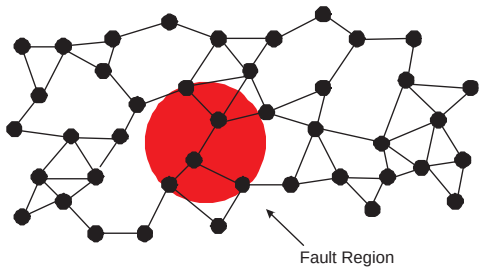


Figure 1.1: Fault region affecting nodes in the communication networks

Question 1: Why general fault tolerant metrics of a network will no longer suffice as a good network designing metric in presence of region-based faults?

Let us get an insight to this question with the help of a of wireless network design scenario. In graph theoretical terms, the *node connectivity* of a graph is the minimum number of nodes that has to be removed before the graph is disconnected. If the network is k-connected, then it will remain connected even after failure of any $k-1$ nodes. Such a network is said to be able to tolerate up to $k-1$ failures. According to region-based failure model, *region-based connectivity* can be defined as the number of nodes that need to fail inside a region to disconnect the whole network [4].

Although the definition of region-based connectivity with single region fault seems to be very close to the definition of node-connectivity of a graph G, quantitatively these two terms may be widely different. Consider the graph shown in Figure 1.2. Clearly, the (node) connectivity of the graph is 2, as failure of the nodes u and v will disconnect the graph. However, if the region is defined to be a subgraph of diameter 2, then the nodes u and v cannot fail simultaneously, as the distance between them is 3 which is greater than the specified diameter of the region. In this situation, the graph will be disconnected if

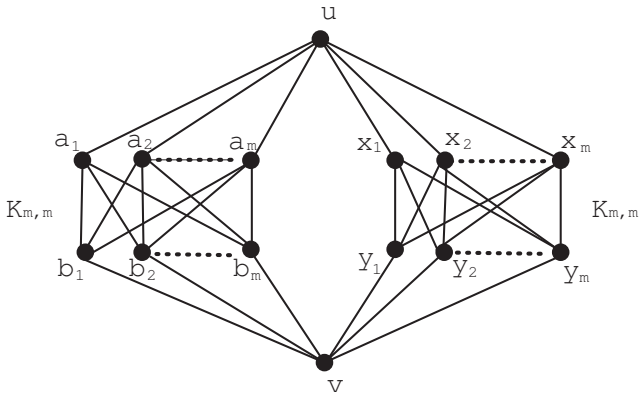


Figure 1.2: A network with connectivity 2 and region-based connectivity $m + 1$

the node set $\{u, b_1, \ldots, b_m\}$ or $\{u, y_1, \ldots, y_m\}$ or $\{v, a_1, \ldots, a_m\}$ or $\{v, x_1, \ldots, x_m\}$ fails. Please note that the distance between any two nodes in any one of the fault sets does not exceed the fault diameter 2. In this case, the minimum number of nodes that has to fail before the graph becomes disconnected is $m + 1$. Accordingly, the region-based connectivity of this graph (with region being a subgraph of diameter 2) is $m + 1$. Since m can be arbitrarily large, the difference between the region-based connectivity and general graph connectivity can also be arbitrarily large. So designing networks using region-based fault tolerant metrics is a whole new problem on its own.

Question 2: What is the advantage of the study of network design considering region-based faults?

Study of network designing techniques, considering region-based faults, is important. If some prior knowledge is available on the maximum size of the region in which faults will be restricted, instead of ignoring that piece of knowledge, it should be utilized for resource efficient design of wireless networks. With the help of the following example of wireless sensor network design scenario, it can

be shown that usage of region-based connectivity leads to a more power efficient design of network over usage of node-connectivity as design parameter.

Suppose that the locations of the sensor nodes are already determined, as shown in the example in Figure 1.3 (the distance between the nodes is shown in the figure). The design requirements specify that the network should be able to tolerate failure of up to 2 sensor nodes. Moreover, it is also given that the faults will only be confined in a single circular region of diameter 5 meters. Since the design requirement specify that the network should be able to tolerate up to 2 faults, one can design a network with connectivity 3. However, in this case, the information regarding the locality of faults (that the faults will be confined to circular region of diameter 5 meters) is ignored. If the network is designed such that the transmission range of each node is 10 meters, it will result in the topology shown in Figure 1.3. It may be noted that the connectivity of this network is 2 as failure of the nodes u and v will disconnect the network. However, the distance between the nodes u and v is 10 meters, and as such they cannot fail simultaneously. Accordingly, this network with each sensor node with transmission range of 10 meters is sufficient to meet to design goal of tolerating up to 2 faults, subject to the condition that the fault will be confined to a circular region of diameter 5 meters. If the information about the locality of faults is ignored, to meet the design requirements, one must design a network whose connectivity is 3. In this example, it implies that the sensor nodes should have sufficient transmission range so that the node b can communicate with node h. This means that the transmission range of the sensors has to be $10\sqrt{3}$ meters instead of 10 meters.

In the above example, the transmission range has to be increased by 71%. Since power consumption is proportional to the square of the transmission

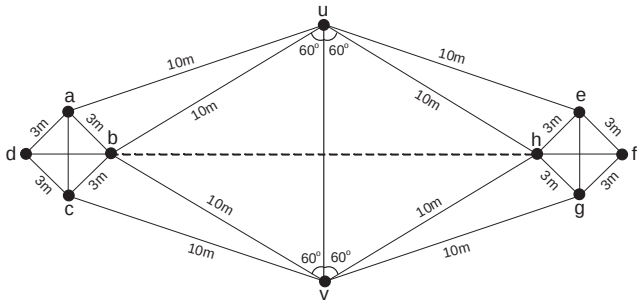


Figure 1.3: An example of a cost effective design using region-based connectivity

range, there has to be 200% increase in power requirements. It may be noted that this increase in transmission power does not provide any additional benefit, because the robustness of the network against failure would have been the same if the transmission power was 10 meters instead of $10\sqrt{3}$ meters. One can conclude form the example that the goal in this case should be to design a network using *region-based connectivity* as a designing parameter instead of *connectivity*. Usage of node-connectivity as fault tolerant metric results in over design of the network with higher costs but no additional benefits.

Even though localized network faults are quite common and important, so far networking research community has contributed very little in the study of different network properties under such fault model.

Implications of considering Region-based Faults in networks

1. **Design of networks**: The knowledge of region-based faults has considerable impact on designing a communication network. If two nodes or links are not within the same region then a single region-based fault will not destroy both of them at the same time. This knowledge will effectively lead to heterogeneous network design. Heterogeneous network design means all nodes in the network need not be strengthen equally in order to maintain the same level of robustness throughout the network.

2. **Resource Efficiency**: This also results in resource efficiency in network design. As shown in previous example, designing a wireless network, with the knowledge of region-based faults, leads to significant reduction of transmission power that is required to gain the same level of network robustness to design a wireless network without this region-based failure information. It has been shown in this thesis, how resource can be efficiently allocated in the networks, *e.g.*, communication networks, data storage networks and data center networks, considering region-based faults.

3. **Analysis of networks**: Analysis of networks will also not remain same if region-based faults are considered instead of general faults. As for example, as shown in this thesis, if region-based faults are considered then minimum number of region-cut and maximum number of region-disjoint paths are no longer equal. Also, study of region-based fault tolerant metrics, *e.g.*, region-based connectivity, region-based component decomposition number etc., are considerably different from the study of general network fault tolerant metrics.

4. **Designs of algorithm for networks**: Network designing algorithms, considering the region-based fault tolerant metrics, are considerably different than network designing algorithms for general fault tolerant metrics. As for example, designing wireless networks with region-based connectivity as design parameter has complexity of $O(n^6 \log n)$ in partial region-fault model, where n is number of nodes in the network. Also, the problems of network design with region-based component number or region-based largest(smallest) component size as design parameters, turn up to be NP-complete and this thesis gives algorithms with performance bounds to design networks with these design parameters.

The principal focus of this thesis is the study of the design and analysis of networks in presence of spatially-correlated faults or region-based faults. First, several new fault-tolerant metrics are proposed along with efficient design techniques suitable for designing communication networks in presence of region-based faults. Also this thesis analyzes the effects of the robustness of these metrics in presence of region-based faults. Secondly, this thesis considers two problems of efficient resource allocation in wide area storage networks and data center networks in presence of region-based faults and presents efficient solutions for them.

1.2 Related Works

Most of the traditional studies on fault-tolerance in wired and wireless networks [5–14] assume that the node/link failures are random in nature, *i.e.*, the probability of a node or link failing is independent of its location. However, the assumption of random node failure is not valid in communication networks in military environment or nature diasters. In these scenarios, faults may be *spatially correlated* or confined in a *region*. The networking research community over last few years has shown interest in studying localized, *i.e.*, *spatially correlated* or *region-based* faults in various types of networks like storage networks [15, 16], overlay networks [17], wide area monitoring services [18], sensor networks [4, 19, 20], content resiliency service networks [21], data center network [22] and fiber-optics networks [23–26].

In order to capture the notion of locality in measuring the fault-tolerance capability of a network, a new variant of connectivity metric called *region-based connectivity* was first introduced in [4]. Also Sen *et al.*, [4] showed that using region-based connectivity over general connectivity definition, one can save

significant amount of transmission power in wireless networks. Region-based connectivity for multiple spatially correlated faults has been studied in [20]. The *region-based connectivity* of a network can be informally defined to be the minimum number of nodes (links) that has to fail within any *region* of the network before it is disconnected.

Research works closely related to network designing problems discussed in this thesis are [19, 23, 25, 27–31]. All these works assess the vulnerability of communication networks under *region-based faults*. In [25], Agarwal *et al.*, take a probabilistic geographically correlated failure into account in measuring average all-terminal reliability and capacity degradations both in terms of the number of failed components and the amount of capacity that is affected. In [19], Liu *et al.*, presented a reliability assessment of wireless mesh networks under probabilistic fault model. The common feature of the reliability assessment of networks, as presented in [19, 25, 27], is that all of these works considered probabilistic region-failure model, *i.e.*, failure probability of a network component tends to monotonously decrease as it is farther away from epicenter of attack. This thesis proposes a deterministic region-failure model for wired and wireless networks which states that, given a region all nodes within the region has equal probability of failure whereas nodes outside the region have zero failure probability.

Neumayer *et al.* [23, 29], on the other hand, considered a single deterministic region-failure model and gave an analysis on identifying the most vulnerable parts of the communication network when the faults are geographically correlated. That is, the analysis gives locations of disasters that would have the maximum disruptive effect on the network in terms of capacity and connectivity. This thesis considers similar deterministic failure model but both with single and

multiple region-faults. In [28,31], authors studied the problem of multiple region-failure problem for planar graphs and gave a generalized result for max-flow and min cut under this scenario.

1.3 Thesis Contributions

As it has been shown in previous sections, region-based faults are quite different from random network faults. Also considering region-based faults, leads to resource efficient design of networks and resource efficient resource allocation in networks. So design and analysis of communication networks in presence of region-based faults is important and needs special attention. This section discusses network designing and resource allocation problems in wireless networks, general networks, data storage networks and data center networks in presence of *region-based faults* and contribution of this thesis in each of them.

1.3.1 Connectivity of Wireless Sensor Networks in Region-Based Fault Model

The studies in fault-tolerance wireless networks mostly focus on the *connectivity* of the graph as metric of fault-tolerance. If the underlying communication network is k-connected, it can tolerate up to $k-1$ failures. In measuring fault tolerance in terms of connectivity, no assumption regarding the locations of the faulty sensors is made - the failed nodes may be very close to each other or very far from each other. In other words, the connectivity metric fails to capture any notion of *locality* of faults. However, wireless networks in open area are exposed to potential threats, *e.g.*, natural disasters (like earthquake, flooding etc) or malicious attacks (EMP attack, bomb explosion), where faults are localized within a certain *region*.

Contributions:

- In order to capture the notion of locality, in this thesis, a new metric called *region-based connectivity* is introduced for measurement of fault-tolerance capability of wireless networks [4, 20]. Region-based connectivity metric is introduced for for both single and multiple fault regions under two fault models - Partial region faults (all nodes inside region need not fail) and Complete region fault (all nodes within a region fail). The attractive feature of the region-based connectivity as the metric is that it can achieve the same level of fault-tolerance as the metric connectivity, but with much lower transmission power for the sensor nodes. This thesis provides both analytical as well as extensive simulation results to support this claim.

- Also this thesis gives an analysis of probability of connectivity of wireless networks in presence a single region-fault under two different wireless signal propagation models - (i) *unit-disk graph model* [32] and (ii) *log-normal shadow fading model* [33].

1.3.2 Design of Networks based on Region Based Components

Robustness or fault-tolerance capability of a network is an important design parameter in both wired and wireless networks. *Connectivity* of a network is traditionally considered to be the primary metric for evaluation of its fault-tolerance capability. However, connectivity $\kappa(G)$ (for *random* faults) or region-based connectivity $\kappa_R(G)$ (for *spatially correlated* or *region-based* faults, where the faults are confined to a region R) of a network G, does not provide any information about the network state, (*i.e.*, whether the network is connected or not) once the number of faults exceeds $\kappa(G)$ or $\kappa_R(G)$. If the number of faults exceeds $\kappa(G)$

or $\kappa_R(G)$, one would like to know, (i) the *number of connected components* into which G decomposes, (ii) the *size of the largest connected component*, (iii) the *size of the smallest connected component*.

Contributions:

- In this thesis, a set of new metrics that computes these values are introduced for scenarios where faults are localized within a certain region. The metrics are - i) *region-based component decomposition number (RBCDN)*, that measures the number of connected components in which the network decomposes once all the nodes of a region fail, ii) *region-based largest component size (RBLCS)*, that measures the the size of the largest connected component and iii) *region-based smallest component size (RBSCS)*, that measures the the size of the smallest connected component after a network gets disconnected due to region-based fault. This thesis proposes an algorithm of polynomial time computational complexity of finding these metrics of a network [34, 35].

- In addition, in this thesis, the problems of least cost design of a network with a target value of RBCDN and RBLCS are studied. The optimal design problem is shown to be NP-complete and efficient approximation algorithms with a performance bound or heuristics are presented. The performance of the algorithms are evaluated by comparing it with the performance of the optimal solution. Experimental results demonstrate that these algorithms produces near optimal solution in a fraction of time needed to find an optimal solution.

1.3.3 Data Distribution Scheme in Data Storage Networks in presence of Region-based Faults

Distributed storage of data files in different nodes of a network enhances the reliability of the data by offering protection against node failure. In the $(\mathcal{N}, \mathcal{K}), \mathcal{N} \geq \mathcal{K}$ file distribution scheme, from a file F of size $|F|$, $\mathcal{N}$ segments of size $|F|/\mathcal{K}$ are created in such a way that it is possible to reconstruct the entire file, just by accessing any $\mathcal{K}$ segments. For the reconstruction scheme to work it is essential that the $\mathcal{K}$ segments of the file are stored in nodes that are connected in the network. However in case of *region-based* node failures the network might become disconnected (*i.e.*, split into several connected components).

Contributions:

- In this thesis, a novel file segment distribution scheme is proposed so that, even if the network becomes disconnected due to any region fault, at least one of the largest connected components will have at least $\mathcal{K}$ distinct file segments with which to reconstruct the entire file. The distribution scheme will also ensure that the total storage requirement is minimized [16]. An optimal solution of this problem is presented through Integer Linear Programming and an approximation solution with a guaranteed performance bound of $O(\ln n)$ is proposed to solve the problem for any arbitrary network. The performance of the approximation algorithm is evaluated by simulation on two real networks. The simulation results show that the approximation algorithm almost always produces near optimal solution in a fraction of time needed to find the optimal solution.

1.3.4 Resource Allocation problem in Data Centers in presence of Region-based Faults

Cloud computing is becoming more relevant and important in current day context as organizations around the world have started migrating their services and applications to large-scale data center infrastructures. A large-scale data center experiences random failures of several hardware components every day. These hardware failures present challenging issues in providing reliable service to the end users. The notion of *fault domains* captures the effect of single hardware failures in data centers. A *fault domain* is defined as a set of servers (correspondingly, the set of VMs hosted on them), all of which become unavailable when a single fault occurs in the data center. For instance, if a switch or a router of the data center network fails, then all the servers connected through this switch/router may become inaccessible. The net effect of failure of such a switch/router is equivalent to simultaneous failure of all the servers connected through this switch/router. Accordingly, this set of servers form a single fault domain.

Contributions:

- In this thesis, the concept of *fault domains* is used and the problem of revenue maximization in fault-tolerant resource allocation in large data centers is investigated [22]. A novel formulation of the problem is presented and proved that this problem is NP-complete. An optimal solution technique is provided through Integer Linear Program formulation and an efficient heuristic is presented that produces near-optimal solution in a fraction of time required to compute the optimal. Through extensive experimentation, the efficacy of the heuristics is presented.

The rest of thesis is organized as follows: Next Chapter 2, formally defines the region-based fault model used throughout this dissertation. In Chapter 3, several problems have been discussed on wireless connectivity in single-region fault model and multiple-region fault model. Chapter 4, defines few new metrics for measuring fault-tolerance capability of a network under region-based fault model. In Chapter 5, this thesis presents an efficient region-based fault-tolerant data distribution scheme in data storage networks. In Chapter 6, an efficient fault-tolerant resource allocation technique in Data Centers is discussed. Finally Chapter 7, concludes the thesis with proposing future direction of this research.

Chapter 2

REGION-BASED FAULT MODEL

The notion of *region-based faults* is tied to the notion of a *region*. Consider a set of nodes distributed over a geographical area. These nodes form a network through wired or wireless links. *Network graph* implies the *topological* relationship between the nodes. In addition to the network graph, there can be a *layout* of the nodes and links in the geographical area.In this thesis the *layout* of the nodes and links is refereed as the *network geometry*. A *region* may be defined either with reference to the network topology or with reference to the network geometry.

- Diameter-based region: A diameter-based region in a network graph $G = (V, E)$ for a given d is defined as a *maximal* induced subgraph of G with diameter[1] d.

- Radius-based region: A radius-based region in a network graph $G = (V, E)$ for a given r is defined as a *maximal* induced subgraph of G containing node v and its r-hop neighborhood. Node v is called as the center of such a region.

- Geometry-based region: With reference to the network geometry, a geometry-based region for a given r is defined as a collection of nodes covered by a circular area of radius r in the network layout. The locations of the nodes are given in a 2-dimensional Euclidean space.

[1] The diameter of a graph is the longest shortest path between any two graph nodes

In this thesis geometry based region definition has been followed in most of the cases. Accordingly some insight on calculating all the regions (geometry based circular regions) on a two dimensional plane is important. Next an analysis on calculating all such regions in polynomial time is given. Input assumptions used for the analysis are as follows:

(i) a network is a graph $G = (V, E)$ where $V = \{v_1, \ldots, v_n\}$ and $E = \{e_1, \ldots, e_m\}$ are the sets of nodes and links respectively, (ii) the layout of G on a 2-dimensional plane $LG = (P, L)$ where $P = \{p_1, \ldots, p_n\}$ and $L = \{l_1, \ldots, l_m\}$ are the sets of points and straight lines on the 2-dimensional plane (note: (a) there is a one-to-one correspondence between the nodes and points in V and P, (b) a one-to-one correspondence between the edges and lines in E and L, (c) each l_i, $1 \leq i \leq m$ connects two points p_j and p_k in P and does not pass through a third point p_q), (iii) a region is defined as a circular area R of radius r, (iv) all the nodes and links inside the fault region are faulty.

In a wired network, a physical link connects two nodes. If a node is destroyed due to failure of a region, all links incident on that node are also destroyed. However, it is possible that failure of a region destroys a link without destroying the nodes at its end points. In a wireless network, there is no physical link, and as such the possibility of a fault destroying a link does not arise. There could potentially be infinite number of circular regions that covers the 2-dimensional plane where the nodes and links are deployed. It may be noted that a node corresponds to a point in this plane and a link (*i.e.*, a straight line in the plane) and a region (*i.e.*, a circular area in the plane) correspond to a *set of points* in the plane. A region R is said to *intersect* line l_i, if $R \cap l_i \neq \emptyset$. Although there could be an infinite number of circular regions in the plane, one only need to consider a finite number of them in order to compute region-based fault tol-

erant metrics of a network. Two regions are said to be *indistinguishable* if they cover the same set of links and nodes. Otherwise, they are *distinguishable* or *distinct*. For computing *region-based fault tolerant metrics* of a network, we only need to evaluate the distinct regions. Since there are n nodes and m links in the network, there could be at most 2^{n+m} distinct regions. It will be shown next that the number of distinct regions that needs to be considered is bounded by a polynomial function of n and m. Two indistinguishable regions are shown in Figure 2.1.

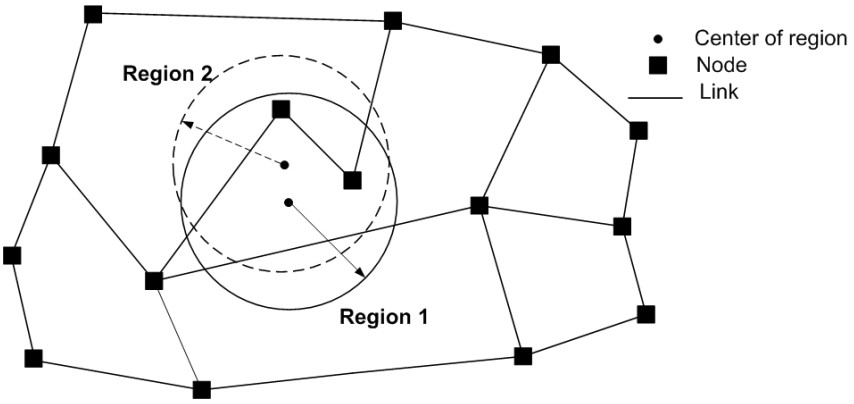


Figure 2.1: Region 1 and Region 2 are indistinguishable

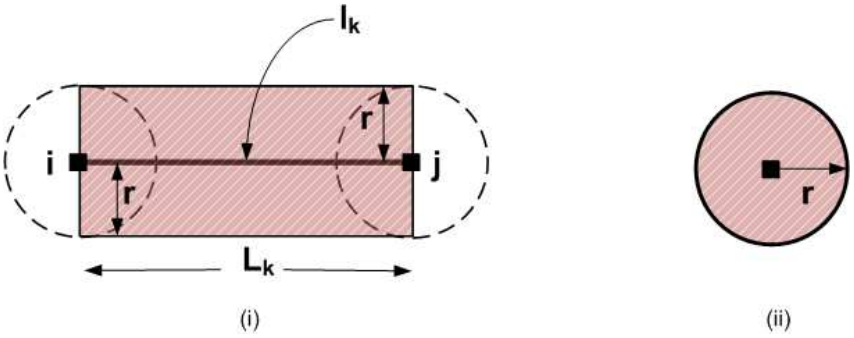


Figure 2.2: Vulnerability zone of a link and a node

Definition 1. Node Vulnerability Zone (NVZ): *The circular area of radius r centered at the location of a node in the network layout is defined as the* NVZ *(see Figure 2.2(ii)). Any region fault occurring in this area will destroy the node.*

Definition 2. Link Vulnerability Zone (LVZ): *Let l_k be a line of length L_k in network layout corresponding to link e_{ij} in the graph. The rectangular area of length L_k and width $2r$ (as shown in Figure 2.2(i)) is defined as the* LVZ *for this link. This area is called LVZ because if the center of the fault region lies within this area, it will destroy the link. It may be noted that a link can also be destroyed if the center of the fault region lies within the NVZ of a node on which the link is incident. However, this area need not be included as part of LVZ, as it is already considered as part of NVZ.*

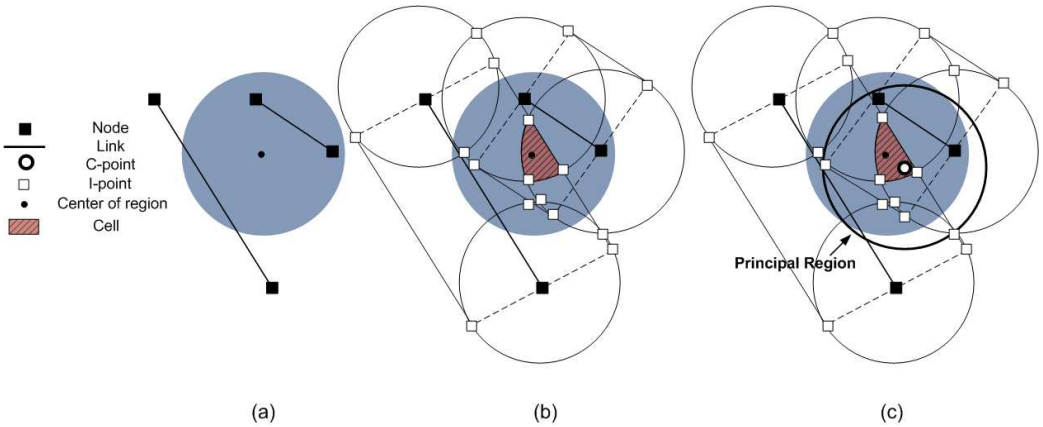


Figure 2.3: A Region, arrangements of vulnerability zones and a Principal Region

Each node and link vulnerability zone can be represented by a set of polynomials $\mathcal{P} = \{P_1, \ldots, P_s\}$ in $\mathbb{R}^2$ with degree(d) at most 2 [36, 37]. Note that number of such polynomials s is $O(n+m)$. The *arrangement* $\mathcal{A}(\mathcal{P})$ [36,37] of the polynomials $\mathcal{P}$ is the subdivision of the plane into *I-points i.e.*,*vertices*, *arcs* and *cells*, where *I-points* are the intersection points of the boundaries of node and link venerability zones [4], and the arcs are the maximally connected portions of the boundaries between the *I-points* and cells are the maximally connected regions bounded by the arcs [36, 37]. The cells can be described by a constant number of polynomial inequalities of constant maximum degree $d = 2$ [36, 37].

In Figure 2.3(a), a region that covers 2 nodes and 2 links (at least partially) is shown. Figure 2.3(b) shows the arrangement of the vulnerability zones of these set of nodes and the links. A cell is highlighted in Figure 2.3(b).

Definition 3. C-point: *A* C-point *is an arbitrarily selected point within a cell in* $\mathcal{A}(\mathcal{P})$*. For each cell only one* C-point *is considered.*

Definition 4. Principal Regions: *Any region centered at a* C-point *will be referred to as a* Principal Region *(Figure 2.3(c)).*

Observation 1. *Given a region* R*, the intersection area of the vulnerability zones of the nodes and links within region* R *is non-empty [4].*

Observation 2. *If a region* R *covers a set of nodes and links and* R *is not centered at one of the* C-points*, there must be at least one other region centered at one of the* C-points *that covers all the nodes and links covered by* R*. Accordingly this region will be indistinguishable from* R*.*

The proof follows the proof of *Observation 2* in [4].

Observation 3. *For computing the* region-based fault tolerant metrics *of the network* G *where the layout* LG *of* G *is given as input, only a limited number of distinct regions, i.e., only the Principal Regions need to be examined [4].*

Observation 4. *The maximum number of Principal Regions is* $O((n+m)^2)$*.*

Proof. By definition, a Principal Region is a region centered at a *C-point* and number of *C-points* is equal to the number of cells in the arrangement $\mathcal{A}(\mathcal{P})$. As

the maximum degree of the set of polynomials $\mathcal{P} = \{P_1, \ldots, P_s\}$ defined over $\mathbb{R}^2$ is 2 [36,37], the number of cells in $\mathcal{A}(\mathcal{P})$ is $O(s^2)$ (where $s = |\mathcal{P}|$). Since s is $O(n+m)$, there can be at most $O((n+m)^2)$ *Principal Regions*. □

Observation 5. *All the* C-points *can be computed in* $O((n+m)\log(n+m) + (n+m)^2)$ *or* $O((n+m))^2)$ *time.*

Proof. If χ is the the total number of vertices, arcs, and faces in $\mathcal{A}(P)$, then using the results presented in [25,36,37], we can compute a set of points $\mathcal{C}$ such that each cell in $\mathcal{A}(\mathcal{P})$ contains at least one point from $\mathcal{C}$, in time $O(s\log(s) + \chi)$ (where $s = |\mathcal{P}| = (n+m)$). As a consequence the overall time-complexity to compute all the *C-points* is

$$O((n+m)\log(n+m) + (n+m)^2)$$

(or $O((n+m)^2)$) as $\chi = O(s^2)$. □

This analysis shows that given a fault radius, number of distinct circular geometrical (or geographical) fault regions is bounded by polynomial factor of number of nodes and links present in the network. Also these distinct regions can be computed in polynomial time. This result has been used throughout the thesis.

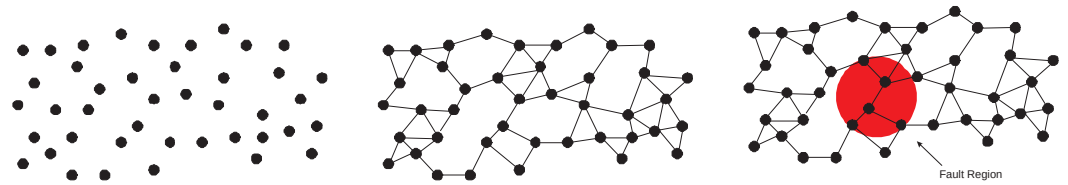
Chapter 3

CONNECTIVITY OF WIRELESS SENSOR NETWORKS IN REGION-BASED FAULT MODEL

A wide area wireless sensor network (WSN) is composed of a large number of sensor nodes distributed in a geographical region. Each sensor node, through its sensing component, gathers information from the surrounding area and then transfer the gathered information to the appropriate location for processing. The sensor nodes in WSN forms an adhoc network among themselves through the communication components, *i.e.*, transmitter and receiver (transceiver). Wide area multi-hop wireless network also works in same way, except it is more concentrated in forwarding communication packets from one part of the network to the other. WSN and wireless networks have gained huge importance in research communities in the last few decades due to its wide potential in civil, military and environmental applications for examples health monitoring, object tracking, intrusion detection, environmental monitoring, disaster recovery, etc [38].

The sensor network should always remain connected so that the nodes can communicate for data fusion and send gathered data to the central data collection point (the base station or the gateway node). In a large WSN, there is always the risk of the sensor nodes getting exposed to unexpected environmental changes and malicious attack. In such cases it is difficult to guarantee that every node in the network to work normally. If any failure take place in WSN then the performance of the network degrades significantly. So among various factors, such as *scalability, fault-tolerance, production cost*, etc that impact the design of sensor networks, fault-tolerance plays a very important role. In graph

theoretical terms, the *connectivity* (node/link) of a graph is the minimum number of nodes (links) that has to be removed before the graph is disconnected. If the sensor network is k-*connected* or k-*link-connected*, then it will remain connected even after failure of any k-1 nodes or links. Such a network is said to be able to *tolerate* up to $k-1$ failures.



(a) Sensor Nodes in the Plane (b) Sensor Network (c) Sensor Network with Fault Region

Figure 3.1: (a) Sensors in a geographic area (b) Communication network formed by the sensors (c) Fault region and the affected sensors

That means by designing a k-connected network topology, it is possible to make a WSN robust against $k-1$ nodes/links failures. Although such designs ensure that the network remains *connected* after the failure of some nodes and links, they fail to capture the information about the location of the faulty nodes. In most of the practical WSNs, faults are most likely to be *massive* in nature and confined within a *limited area or region*. Some examples of such massive but localized faults in wireless sensor networks are as follows: (i) due to natural calamities like earthquake, forest fire or flood disaster, environment monitoring sensor devices of a particular region can get destroyed, (ii) sensor nodes in a certain part of the network may die due to energy depletion for heavy traffic [2], (iii) by performing jamming attack on the network a jammer may create a *jamming hole* in a certain part of the network making the nodes in that part inoperable [3], (iv) similarly in a military environment EMP attack or an enemy bomb can destroy large number of sensor devices within in a particular area of

the network. Such situations are shown in Figure 3.1b, where the shaded region shows the affected region.

In order to capture the notion of locality of fault-tolerance capability of the network, a new notion of connectivity called *region-based connectivity (RBC)* was introduced by Sen *et al.*, [4]. This research discusses about *region-based connectivity* in two different fault models (i) partial region fault model (PRFM), where some nodes within a region of WSN can fail and (ii) complete region fault model (CRFM), where all the nodes inside the region will fail. Again, region-based failures in wireless sensor networks can be of two types - (i) *single region fault* where faults are confined to one region only and (ii) *multiple region fault* where faults are confined to k regions for some specified integer k. The results presented in [4] are limited to RBC in PRFM for single region fault only. In next section, an extension of Sen *et al.*'s work [4] is given for multiple region fault in PRFM. Also, this thesis studies region-based connectivity for multiple region-based faults in CRFM. In addition, in subsequent sections, this thesis provides a probabilistic analysis of the connectivity of wireless sensor networks using (i) unit disk graph model and (ii) log-normal shadow fading model, as signal propagation model, in presence of a single region fault.

3.1 Connectivity of Wireless Sensor Networks in Partial Region Fault Model (PRFM)

In PRFM, the *partial-region-based connectivity* $p\kappa_R^k(G)$ of a graph G with region R and for a particular value of k, is the minimum number of nodes that need to fail from any k-regions so that the graph G gets disconnected. This is called *partial-region fault model* since all the nodes in the fault region may not fail simultaneously. In special case where $k = 1$, $p\kappa_R(G)$ can be formally defined as follows:

Definition 5. Partial-region-based connectivity*: Suppose that {$R_1, \ldots, R_l$} is the set of all possible regions of the graph G. Consider a l-dimensional vector T whose i-th entry, $T[i]$, indicates the number of nodes in region R_i whose failure will disconnect the graph G. If the graph G remains connected even after the failure of all nodes of the region R_i, then $T[i]$ is set equal to ∞.*
The partial-region-based connectivity *of a graph G with region R is* $p\kappa_R(G) = \min_{1 \leq i \leq k} T[i]$.

In [4] Sen *et al.*, proved that given a layout of a sensor network of n nodes on a two dimensional plane and a region radius r, the total number of principle regions that need to be considered in that deployment area in PRFM is of the $O(n^2)$. Also the authors gave two algorithms of polynomial time complexity to find the *partial-region-based connectivity* $p\kappa_R^k(G)$ of the network under any single region fault. Sen *et al.*'s work also developed an algorithm of polynomial time complexity, $O(n^6 log\ n)$ in PRFM when $k = 1$, for the design of a sensor network with region-based connectivity $K = p\kappa_R^k(G)$ with minimum transmission power to the sensor nodes.

A simple extension of the network design algorithm for multiple region faults, *i.e.*, for $k > 1$, can be given in PRFM. The time complexity of the algorithm then increases to $O(n^{2k+4} log\ n)$ but still remains of polynomial with respect to the number of nodes n in the network and number of fault regions k.

In Chapter 1, it was demonstrated with a concrete example that substantial savings in power can be realized by using the new metric, region-based connectivity, instead of the conventional metric, connectivity. However, questions remain if such savings in power can be realized in more general cases, or it is true only in some pathological cases. Next, with extensive simulation re-

sults it will be shown that such energy saving is in fact realizable in more general cases.

3.1.1 Results for region-based connectivity problem in PRFM

In this simulation environment, the x and y coordinates of n sensor nodes are uniformly generated in a $1000 \times 1000\ m^2$ field. The transmission range required to ensure the desired Graph Connectivity is compared with the transmission range required to ensure the desired region-based connectivity in PRFM in different experimental settings. The following four parameters can influence the comparison results: (i) the number of nodes n, (ii) the region radius r, (iii) maximum number of regions k where faults occur, and (iv) the desired graph connectivity or the region-based connectivity K. For each of these four parameters, a set of experiments is performed to find its effect on the transmission range.

In the first set of experiments, the effect of changing connectivity requirement K on the transmission range is noted, while the number of nodes n and the region radius r are kept constant. This set of experiments are conducted for two sets of values of n and r. For the first set $n = 100$ and $r = 100\ m$ and for the second set $n = 100$ and $r = 150\ m$. In both these sets of experiments, k is set to 0, 1 and 2 and the connectivity parameter K is changed from 2 to 25. The case with $k = 0$ represents *no-fault scenario* and general graph connectivity is measured instead of region-based connectivity. The results of these two experiments are presented in Figure 3.2. Each experiment is repeated 10 times, and the results are averaged over 10 randomly generated topologies. It may be observed that in Figure 3.2 the transmission range for both connectivity and region-based connectivity increases with the increasing value of the parameter K. However, the rate of increase in transmission power for connectivity is much

higher than the corresponding increase in region-based connectivity. The difference in transmission range (and hence power requirement) for connectivity and region-based connectivity grows with the increasing value of K, implying that there exists greater potential for power saving for larger values of K.

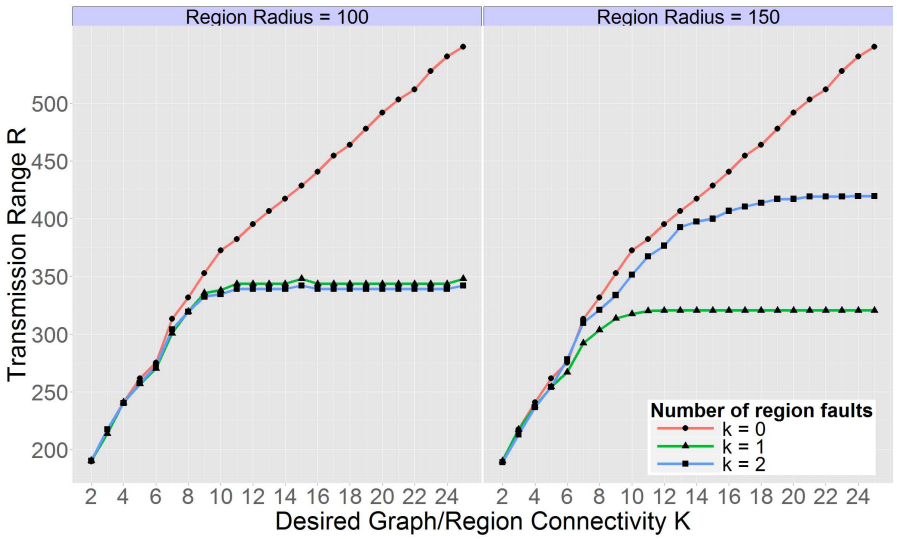


Figure 3.2: Transmission Range vs. Connectivity K, with $n = 100$

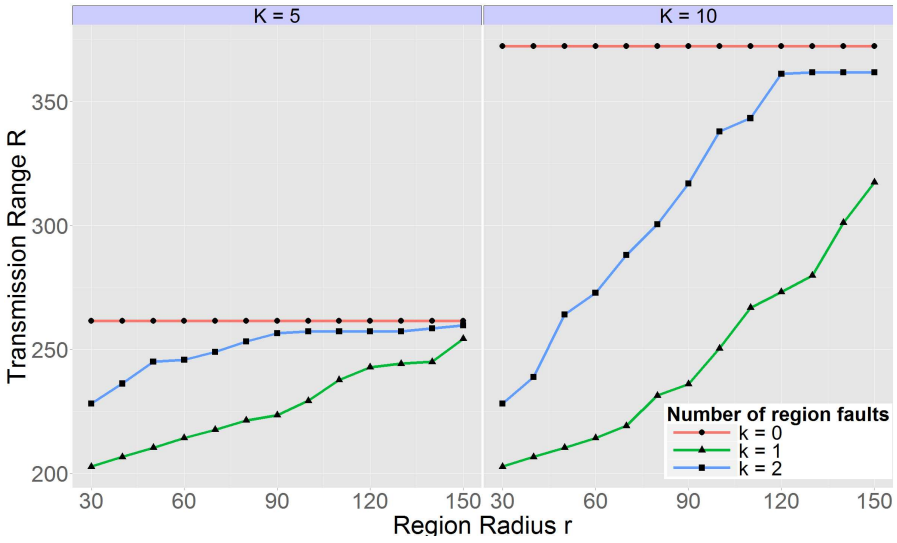


Figure 3.3: Transmission Range vs. Region Radius r, with $n = 100$

Figure 3.2 shows that the transmission range requirement saturates for

region-based connectivity, once the value of K reaches a certain threshold value (12 when $r = 100$ m and 8 when $r = 150$ m for $k = 1$ in Figure 3.2). This saturation effect is observed because the region radius is set equal to 100 m and 150 m in these two sets of experiments. Once the transmission range exceed the value of r, even failure of all the nodes in a region may not disconnect the graph. As a result no further increase in transmission range is needed for region-based connectivity. The region-based connectivity of the graph become ∞ at this time.

In the second set of experiments, the effect of changing the region radius r on the transmission range is noted, while the number of nodes n and the connectivity K are kept constant. This set of experiments are conducted for two sets of values of n and K. For the first set $n = 100$ and $K = 5$ and for the second set $n = 100$ and $K = 10$. In both these sets of experiments, the region radius r is changed from 30 to 150. The results of these two experiments are presented in Figure 3.3.

As in the previous set, each experiment is repeated 10 times, and averaged over 10 randomly generated topologies. It may be observed that the value of the parameter r has practically no effect on the transmission power requirement to maintain a certain graph connectivity ($K = 5$ and $K = 10$ in our experiments when $k = 0$). On the other hand the the parameter r has considerable impact on the transmission range to maintain the specified region-based connectivity 5 and 10. It can be seen from the Figure 3.3, that in order to maintain the specified region-based connectivity, the transmission range has to increase with the region radius r. Both the observations regarding the behavior of connectivity and region-based connectivity are quite expected. As the radius of the region increases, the restriction on faults being localized is also relaxed and as

such transmission range requirement for region-based connectivity tends to approach the transmission range requirement for connectivity. It may be noted that when the radius of the region is sufficiently large, there will not be any difference between the region-based connectivity and graph connectivity.

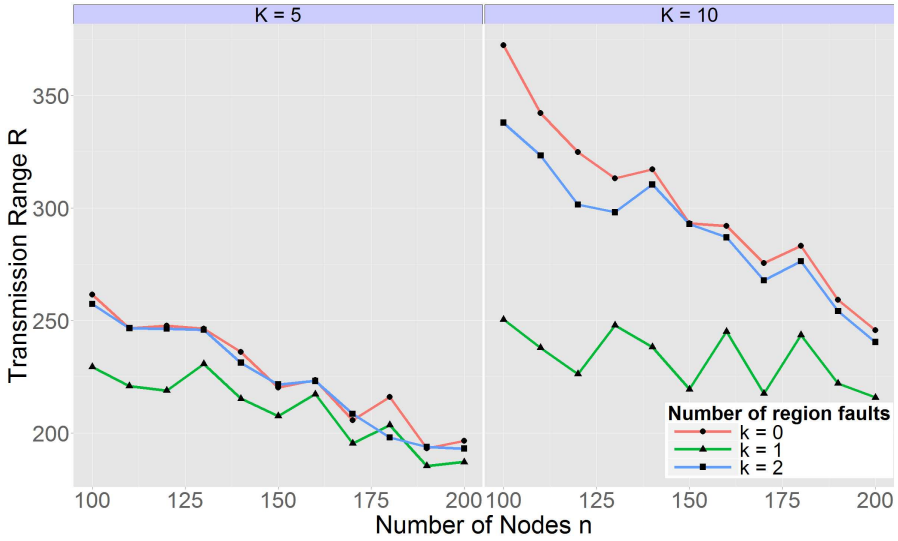


Figure 3.4: Transmission Range vs. Number of Nodes n, with $r = 100$ m

In the third set of experiments, the effect of changing the number of nodes n on the transmission range is noted, while the region radius r and the connectivity K are kept constant. This set experiments was conducted for two sets of values of r and K. For the first set $r = 100$ m and $K = 5$ and for the second set $r = 100$ m and $K = 10$. In both these sets of experiments, the region radius r is changed from 100 m to 200 m. The results of these two experiments are presented in Figure 3.4. As in the previous cases, each experiment is repeated 10 times, and averaged over 10 randomly generated topologies.

It may be observed in Figure 3.4 that both the transmission range to maintain connectivity and region-based connectivity K, decreases with increase in the number of nodes in the sensing field. This is quite expected as the higher

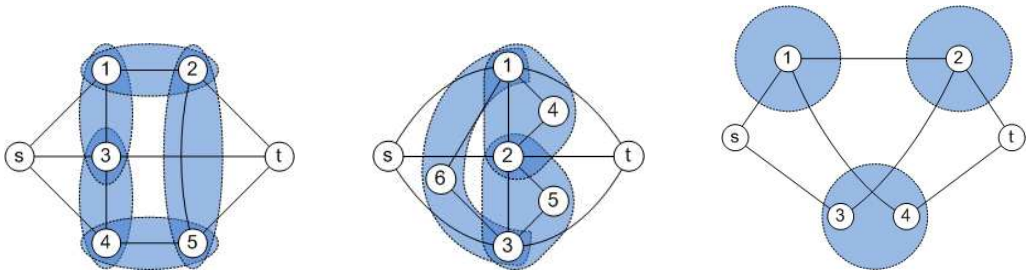
node density implies closer proximity of the nodes and closer proximity allows for the sensor network graph to remain connected, even with shorter transmission range. The difference between transmission range for connectivity and region-based connectivity also decreases with increase in the number of nodes. This is expected because with higher node density, each node v_i will have a larger number of closely located neighbors. These nodes have to fail before v_i gets disconnected from the network. In a sense the failures have to be localized. This is exactly the same situation that the region-based connectivity tries to capture. For this reason, with a large number of nodes in the sensing field, the transmission power requirement for both connectivity and region-based connectivity approaches the same value.

It may be noted that in all of these experiments, the region-based connectivity needed smaller transmission range (and hence lower power requirement) than the connectivity metric. In some situations, the requirements for both the metrics approached the same value, but in no case did the connectivity metric require a smaller transmission range (and smaller power) than required by the metric region-based connectivity. From these sets of experiments one can conclude that in comparison with connectivity as the metric for fault-tolerance, the use of region based connectivity will result in substantial savings in energy for most of the situations.

3.2 Connectivity of Wireless Sensor Networks in Complete Region Fault Model (CRFM)

In CRFM, the *complete-region-based* $c\kappa_R(G)$ *(node*[1]*) connectivity* of graph G with a specified definition of region R is defined as minimum number of regions whose removal (*i.e.*, removal of all nodes in the regions) will disconnect the

[1]The terms *node* and *vertex* are used interchangeably in this thesis.



(a) Diameter-based region definition (Maximal subgraphs of diameter $d = 1$): Set of regions excluding regions containing the source and destination are R_1=$\{1, 2\}$, R_2=$\{1, 3\}$, R_3=$\{3, 4\}$, R_4 =$\{4, 5\}$, R_5=$\{2, 5\}$. Here, $c\kappa_R(G)$=$\{R_2, R_3\}$, but $|\mathcal{P}| = 1$

(b) Radius-based region definition (Maximal subgraphs containing a vertex and its $r = 1$-hop neighborhood): Set of regions excluding regions containing the source and destination are R_1=$\{1, 2, 4\}$, R_2=$\{2, 3, 5\}$, R_3=$\{1, 3, 6\}$, R_4=$\{2, 4, 5\}$. Here, $c\kappa_R(G)$=$\{R_1, R_2\}$, but $|\mathcal{P}| = 1$

(c) Geometry-based region definition (Circular area representing geometric regions): Set of regions excluding regions containing the source and destination are R_1=$\{1\}$, R_2=$\{2\}$, $R_3 = \{3, 4\}$. Here, $c\kappa_R(G)$=$\{R_1, R_2\}$, but $|\mathcal{P}| = 1$

Figure 3.5: Example graphs in which size of minimum region cut $c\kappa_R(G)$ may not be equal to maximum number of region-disjoint paths $\mathcal{P}$ for different definitions of region

graph. It should be noted that concept of region failure in CRFM is somewhat different from PRFM. In PRFM one is interested in the minimum number of nodes, inside any k regions, whose failure will disconnect a graph. While in CRFM, failure of a region means failure of all nodes inside that region.

Analogous to the notion of vertex cut[2] in graphs, a *region cut* of a graph can be defined as a set of regions whose removal disconnects the graph. It is to be noted that the different regions in a graph may overlap with each other. In particular, a node can be part of multiple regions. Let us define the *region set of a node* v denoted by $R(v)$, as the set of regions that node v belongs to. The *region set of a path* P denoted by $R(P)$ is defined as the union of the region sets of the nodes of P. Given two nodes s, t in $G = (V, E)$, two paths P_1 and

[2]A vertex cut of a graph $G = (V, E)$ is a subset of nodes whose removal disconnects G.

P_2 between s and t are called as *region-disjoint paths* if the union of regions associated with the internal nodes of P_1 and the union of regions associated with the internal nodes of P_2 are disjoint. (Internal nodes of P_1 and P_2 are the nodes of P_1 and P_2 excluding the start node s and the termination node t). In other words, P_1 and P_2 are *region-disjoint* if sets $R(P_1) \setminus \{R(s) \cup R(t)\}$ and $R(P_2) \setminus \{R(s) \cup R(t)\}$ are disjoint. As for example in Figure 3.5c there are 3 paths between source s and destination t : $P1 : s \rightarrow 1 \rightarrow 2 \rightarrow t$, $P2 : s \rightarrow 1 \rightarrow 4 \rightarrow t$ and $P3 : s \rightarrow 3 \rightarrow 2 \rightarrow t$. So the regions associated with the paths excluding regions containing the source and destination are $R(P1) = \{R_1, R_2\}$, $R(P2) = \{R_1, R_3\}$ and $R(P3) = \{R_2, R_3\}$. Since none of $R(P1)$, $R(P2)$ and $R(P3)$ are disjoint with each other, we will say there is only *one region-disjoint path*.

One classical result of Graph Theory is *Menger's Theorem (1927)* which states that if u and v are two nodes in a graph $G = (V, E)$, then the *minimum* number of nodes whose removal disconnects u and v in G is equal to the *maximum* number of node-disjoint paths between u and v [39]. With the introduction of the notion of region-disjoint paths in CRFM, a fundamental research question arises: *Does Menger's Theorem hold for region-based connectivity in CRFM?* More specifically the question is: *Is the* minimum *number of* regions *whose removal disconnects u and v in G, equal to the* maximum *number of* region *disjoint paths between u and v*? We show that the answer to this question is negative through a few illustrative examples shown in Figure 3.5.

Claim 1. *For a variety of definitions of a "region", the minimum number of regions whose removal disconnects the nodes s and t in G (*i.e.*, the region cut RC between s and t), is* not *equal to the maximum number of region-disjoint paths (RDP) between s and t. Moreover, the difference between minimum RC and*

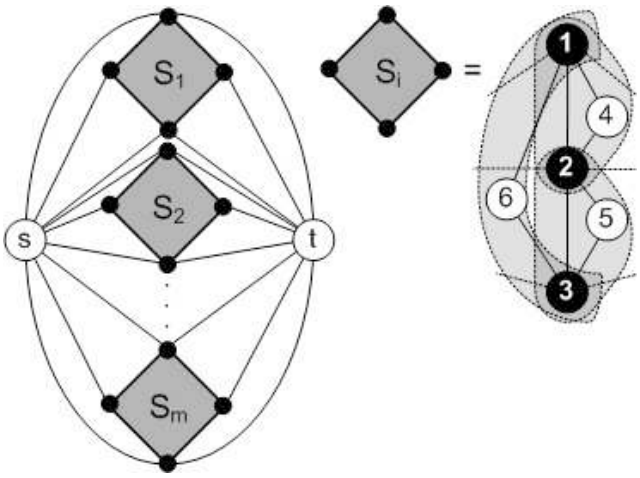


Figure 3.6: A network with RDP and RC equal to m and 2m respectively

maximum RDP between s and t can be arbitrarily large.

Proof. Let $\mathcal{P}$ denote the largest set of region-disjoint paths between a node pair s and t in the graph and let $c\kappa_R(G)$ denotes the size of the minimum region cut. It is trivially true that $c\kappa_R(G) \geq |\mathcal{P}|$ since otherwise, the graph will be connected even after the removal of all the regions in the region cut. In the example graphs of Figure 3.5, the size of minimum region cut is greater than the maximum number of region-disjoint paths between nodes s and t even with three different definitions of a region. In Figures 3.5 (a), (b) and (c), minimum RC is 2 while maximum RDP is 1 between the nodes s and t with three different definitions of a "region". The difference between the minimum RC and maximum RDP in the Figure 3.5(b) is 1. However, by repeating the graph structure of Figure 3.5(b) m times (as shown in Figure 3.6), the difference between minimum RC and maximum RDP can be made as high as m. Accordingly, the difference between minimum RC and maximum RDP between a node pair s and t can be arbitrarily large. □

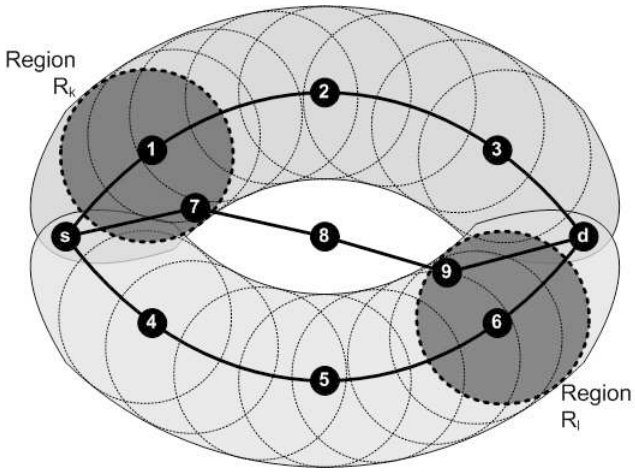


Figure 3.7: A network with 3 node disjoint paths but 2 region disjoint paths

It should be noted that determination of minimum RC and maximum RDP are both equally important for a networks. If a network is k-region fault tolerant then failure of $k-1$ regions in the network will not disconnect the network and there will be at least one path between every pair of remaining nodes. Design specification of a network may require it to be k-region fault tolerant. Region disjoint paths are important for fault tolerant multi-path routing. It should be noted that node disjoint paths in the underlying graph may not be region disjoint. Failure of a region may eliminate two or more node disjoint paths. As for example the network shown in Figure 3.7 has three *node-disjoint paths*, $P1 : s \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow d$, $P2 : s \rightarrow 7 \rightarrow 8 \rightarrow 9 \rightarrow d$ and $P3 : s \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow d$. But out of them only two, $P1$ and $P3$, are *region-disjoint paths*. Since failure of one region may disconnect more than one node disjoint paths, it is more appropriate to find region-disjoint paths than node disjoint paths. In order to design a network with these parameters we need to know how to compute these parameters first.

In next two sections we are mainly concentrating our focus on computation of minimum region connectivity and maximum region disjoint paths between a pair of nodes in a network, in CRFM.

3.2.1 Computation of Minimum Region Cut in CRFM

In this section, formal statement of the Minimum Region Cut (MRC) Problem in CRFM is given along with the proof that the problem is NP-complete and a heuristic for the solution of the problem.

Problem 1. *Given a graph $G = (V, E)$, two vertices $s, t \in V$ and a set of regions*[3] *$\mathcal{R} = \{R_1, R_2, \ldots, R_m\}$, the MRC problem is to compute the smallest subset of regions $\mathcal{R}_s \subseteq \mathcal{R}$ whose removal disconnects s and t.*

The restricted version of MRC problem can be shown to be NP-complete when the regions are computed using the geometry-based region definition.

3.2.1.1 Complexity Analysis of the MRC Problem

The decision version of the restricted MRC problem is as follows.

INSTANCE: Graph $G = (V, E)$, vertices $s, t \in V$, *non-planar* layout of the graph on a plane, a set of circular regions $\mathcal{R} = \{R_1, R_2, \ldots, R_m\}$ of radius r and integer K

QUESTION: Is there a subset $\mathcal{R}_s \subseteq \mathcal{R}$ whose removal disconnects s and t such that $|\mathcal{R}_s| \leq K$?

Theorem 1. *MRC problem is NP-complete [31]. In [31], Bienstock showed that MRC problem can be solved in polynomial time if the graph G is planar. But*

[3] None of the regions in this set contain the vertices s or t, since region cut for s and t will be meaningless if s or t itself is removed.

the problem becomes NP-hard when G is non-planar. An instance of set-cover problem can be reduced to MRC problem when G is non-planar. The nature of the reductions (from set-covering) is such that even polynomial-time approximation algorithms for these problems are unlikely to exist.

3.2.1.2 Solution to the MRC Problem

Since computing minimum region cut in a non-planar graph is NP-complete, this thesis provides an efficient heuristic to compute it. An input to the MRC problem is the set of regions computed with geometry-based region definitions. The region enumeration process is given in Chapter 2.

Heuristic for computing Region Cut

Given a graph $G = (V, E)$, a set of regions $\mathcal{R} = \{R_1, R_2, \ldots, R_m\}$ and two vertices $s, t \in V$, a *simple* but *efficient* heuristic is proposed to compute the minimum region cut for s and t (Algorithm 1). The intuition behind the heuristic is based on the fact that the set of vertices belonging to a region cut of s and t is also vertex cut for s and t. Each vertex in the vertex cut may belong to multiple regions and a region may contain one or more vertices of the vertex cut. For each vertex cut, the set of fewest regions that contains all the vertices of the vertex cut can be found (using a set cover construction procedure explained below). The minimum region cut for s and t is then the smallest among all such sets of region covers. It is to be noted that it is sufficient to consider only the *minimal* vertex cuts, instead of considering *all* vertex cuts in the graph. A *minimal* vertex cut for vertices s, t in a graph G is a vertex cut V_c of s, t such that no proper subset of V_c is a vertex cut of s, t in G.

However, there may be an exponential number of minimal vertex cuts in a graph. Thus, only ρ unique minimal vertex cuts are generated in the heuristic,

where ρ is a configurable parameter of the heuristic. The algorithm given in [40] is used to generate ρ unique minimal vertex cuts. The algorithm in [40] employs a recursive procedure to generate minimal vertex cuts at $O(|V||E|)$ or $O(n^3)$ computational effort per vertex cut.

The heuristic then constructs a set cover instance, in which there exists an element for each vertex of the vertex cut and a subset corresponding to each region in the graph. If a region R_i contains vertices V' of the vertex cut V_c, then the subset corresponding to R_i will contain elements corresponding to vertices V'. Since optimal set cover is NP-complete, a well-known greedy set cover approximation algorithm is applied to solve the set cover instance. The greedy set cover selects in each iteration, the subset S that covers the largest number of remaining elements that are uncovered. This polynomial-time algorithm provides an approximation ratio of $\ln |X|$, where $|X|$ is the number of vertices in the vertex cut [41]. The running time of the greedy set cover algorithm is $O(n)$ [41]. So time complexity of CRC (Algorithm 1) is $O(n^3)$.

Optimal Solution for computing Region Cut

In the heuristic for the MRC problem, there are two places for loss of accuracy of the heuristic. The first is due to the fact that in order to reduce computation time we generate only ρ minimal vertex cuts instead of all minimal vertex cuts. The second place of inaccuracy is the approximate solution to the set cover for each minimal vertex cut. In order to compute the optimal solution for the MRC problem, we set ρ to be a sufficiently large value $(O(2^{|V|}))$ so that all minimal vertex cuts in the graph are generated and solve the set cover instance optimally using an Integer Linear Program formulation

Algorithm 1: Compute Region Cut (CRC)

Input : Graph $G = (V, E)$, set of regions $\mathcal{R} = \{R_1, R_2, \ldots\}$, vertices s and t, heuristic parameter k

Output: region cut $\mathcal{R}_s \subseteq \mathcal{R}$

Initialize $\mathcal{R}_s = \emptyset$;

Generate k *minimal* vertex cuts $\mathcal{V}_c = \{V_1, V_2, \ldots, V_k\}$ in G using the algorithm in [40];

forall the $V_i \in \mathcal{V}_c$ *with* $V_i = \{v_{i_1}, v_{i_2}, \ldots, v_{i_n}\}$ **do**

    $\forall 1 \leq l \leq n$, let R_{i_l} be the set of regions containing vertex v_{i_l}. Let the number of unique regions containing all the vertices of V_i be m;

    Construct set cover instance with $X = \{x_{i_1}, x_{i_2}, \ldots, x_{i_n}\}$ and family of subsets $\mathcal{F} = \{S_1, S_2, \ldots, S_m\}$. For each vertex $v_{i_l} \in V_i$, if $v_{i_l} \in R_{i_j}$ then $x_{i_l} \in S_j$;

    Solve the set cover instance using the greedy set-cover algorithm. Suppose the solution returned by the set-cover algorithm is $\{S_{i_1}, S_{i_2}, \ldots, S_{i_p}\}$ and the regions corresponding to these subsets is $\{R_{i_1}, R_{i_2}, \ldots, R_{i_p}\}$;

    if $|\mathcal{R}_s| < p$ **then**

        $\mathcal{R}_s \leftarrow \{R_{i_1}, R_{i_2}, \ldots, R_{i_p}\}$;

return $\mathcal{R}_s$;

It may be noted that the complete-region-based connectivity $c\kappa_R(G)$ of the graph G can be obtained by invoking region cut algorithm on all vertex pairs in the graph and taking the minimum of the region cut values. Accordingly, depending on whether exact $c\kappa_R(G)$ is required or approximate $c\kappa_R(G)$ is sufficient, the optimal or heuristic region cut algorithms can be applied respectively.

3.2.2 Computation of Maximum Number of Region-Disjoint Paths in MRFM

In this section, the Maximum Number of Region-Disjoint Paths (MRDP) Problem in CRFM is formally stated and also proved that the problem is NP-complete and a heuristic for the solution of the problem is given.

Problem 2. *Given a graph $G = (V, E)$, layout of the graph on a plane, two vertices $s, t \in V$ and a set of regions $\mathcal{R} = \{R_1, R_2, \ldots, R_m\}$, the MRDP problem is to compute maximum number of region-disjoint paths between s and t.*

3.2.2.1 Complexity Analysis of the MRDP Problem

A restricted version of MRDP problem is NP-complete.

INSTANCE: Graph $G = (V, E)$, vertices $s, t \in V$, *non-planar* layout of the graph on a plane and a set of circular regions $\mathcal{R} = \{R_1, R_2, \ldots, R_m\}$ of radius r

QUESTION: Are there *two* region-disjoint paths between s and t?

Theorem 2. *MRDP problem is NP-complete [31].*

Proof. A *hole* is a set of vertices in a graph G that induces a chordless cycle in G. The *hole recognition* problem is to determine if a hole passing through two vertices in a graph exists. This problem was proved to be NP-complete in [31]. In the diameter-based definition of region with $d = 1$, two paths will be region-disjoint if and only if they induce a chordless cycle. Thus, for $d = 1$, the restricted MRDP problem is equivalent to the *hole recognition* problem in a graph. □

3.2.2.2 Solution to the MRDP Problem

Given the graph $G = (V, E)$, vertices $s, t \in V$ and set of regions $\mathcal{R} = \{R_1, R_2, \ldots, R_m\}$, the MRDP heuristic computes many region-disjoint paths between s and t (Algorithm 2). The heuristic works in two phases. In the first phase, it computes maximum number of node-disjoint paths between s and t in the graph using max-flow-min-cut techniques [41]. If the computed node-disjoint paths are also pairwise region-disjoint then we are done. If not,

the heuristic enters the second phase in which it finds a large subset of the node-disjoint paths that is region-disjoint. This is accomplished by constructing a ***path intersection graph***. Suppose $\mathcal{P}_{ND} = \{P_1, P_2, \ldots, P_n\}$ are the n node-disjoint paths computed in the first phase. In the second phase, the heuristic constructs the *path intersection graph* $G' = (V', E')$, in which a vertex $v_i \in V'$ corresponding to a path $P_i \in \mathcal{P}_{ND}$. If two paths $P_i, P_j \in \mathcal{P}_{ND}$ pass through at least one common region, then there will be an edge $(v_i, v_j) \in E'$. In the path intersection graph G', an independent set of vertices corresponds to a set of region-disjoint paths in graph G and hence, the heuristic finds a large independent set of vertices in G'. Since the problem of finding maximum independent set in a graph is NP-complete [42], the heuristic uses a greedy approach to find an independent set in G'. The greedy algorithm in each iteration selects the vertex with the minimum degree into the independent set and removes the vertex and its neighbors from the graph. The procedure is repeated until there are no vertices left in the graph. The heuristic returns the region-disjoint paths corresponding to the vertices in the independent set.

Algorithm 2: Compute Region Disjoint Paths (CRDP)

Input : Graph $G = (V, E)$, set of regions $\mathcal{R} = \{R_1, R_2, \ldots\}$, vertices s and t
Output: Set $\mathcal{P}$ of region-disjoint paths

1 $\mathcal{P} \leftarrow \emptyset$;
2 Compute $\mathcal{P}_{ND} = \{P_1, P_2, \ldots, P_n\}$, the set of node-disjoint paths between s and t in G;
3 Construct *path intersection graph* $G' = (V', E')$, where $\forall P_i \in \mathcal{P}_{ND}, v_i \in V'$ and edge $(v_i, v_j) \in E'$ if paths P_i, P_j are not region-disjoint;
4 **while** $V' \neq \emptyset$ **do**
5 Find vertex $v_i \in V'$ such that v_i has the smallest degree;
6 $\mathcal{P} \leftarrow \mathcal{P} \cup \{P_i\}$;
7 $V' \leftarrow V' \setminus \{v_i \cup N(v_i)\}$, where $N(v_i)$ is the set of neighbors of v_i;
8 **return** $\mathcal{P}$;

Optimal Solution for computing Region-Disjoint Paths

In order to obtain the optimal solution to the MRDP problem, first all possible node-disjoint paths between s, t is enumerated using max-flow-min cut techniques [41]. Next, the *path intersection graph* is constructed on the set of all the paths. An optimal independent set of the path intersection graph using Integer Linear Program (ILP) formulation.

3.2.3 Experimental Results and Discussion

In order to evaluate the effectiveness of the proposed heuristics, extensive simulations are conducted. In these experiments, the results obtained from the heuristic were compared with the optimal value of the region cut. The optimal solutions were obtained by solving the Integer Linear Programs using CPLEX Optimizer 10.0. In all these experiments, the location coordinates for the n nodes of the graphs are uniformly generated in a layout of $1000 \times 1000\ m^2$. We consider three parameters that impact the comparison results: (i) the number of nodes n, (ii) the region radius r and (iii) the transmission range T_r of the nodes. A number of experiments were conducted to measure the effect of these parameters on the region cut and number of region-disjoint paths in the graph.

3.2.3.1 Results for the MRC Problem

It may be noted that the heuristic to compute RC (Algorithm 1) starts with a number of minimal vertex cuts ρ. In the experiments this number (ρ) was taken to be 50. In the first set of experiments, we set region radius $r = 100$ m and vary the transmission range T_r of the sensor nodes from 250 m to 500 m in steps of 50 m. For each transmission range, the difference between the region cut produced by the heuristic and the optimal for *all source-destination pairs* in

the graph is measured. Experiments are performed for graphs having different number of nodes (20, 30 and 40 nodes). For each value of n, 10 random graphs were generated each having n nodes. The results were averaged over the 10 graphs. Figure 3.8 shows the percentage of cases where RC computed by the heuristic differed from the optimal cut value by i for $0 \leq i \leq 4$ and $i > 4$. It may be observed that in all the three types of graphs, the heuristic computed the optimal region cut for more than 92% of the source-destination pairs.

The results of experiments showing the effect of the transmission range on region connectivity is shown in Figure 3.9. This set of experiments was performed on graph of 30 nodes and region radius 100. It may be observed that the region connectivity increases with the increase in transmission range. This is natural since increase in transmission range induces more edges in the graph, resulting in increased connectivity. Thus, more regions have to be removed on an average to disconnect two nodes in the graph. As in the previous case, the heuristic gives near-optimal results. In the third set of experiments on region connectivity, the region radius r is varied from 50 m to 110 m in steps of 20 on a randomly generated graphs of size 30 while keeping the transmission range T_r constant at 150 m. The results of the experiment are shown in Figure 3.10. As seen in the figure, the region connectivity decreases with the increase in region radius. This is because when region radius is increased, a region may contain more nodes than before and therefore fewer regions need to be removed before the source destination node pair get disconnected. Here again, the heuristic gives near-optimal results.

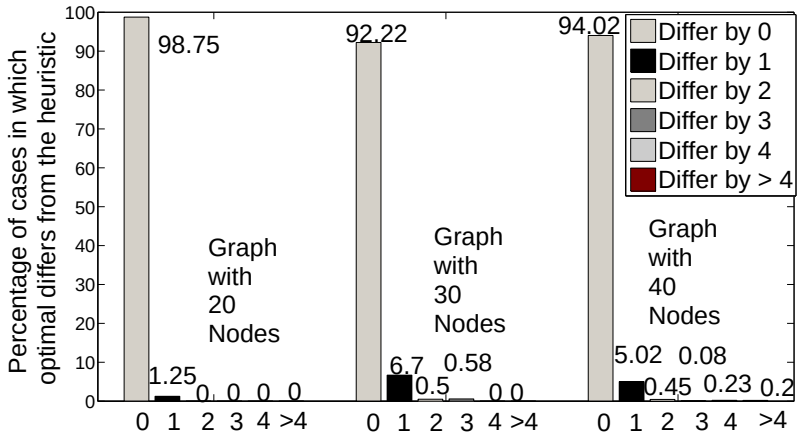


Figure 3.8: Percentage of cases in which region cut computed by heuristic differs from optimal value of region cut for three different random graphs of size 20, 30 and 40

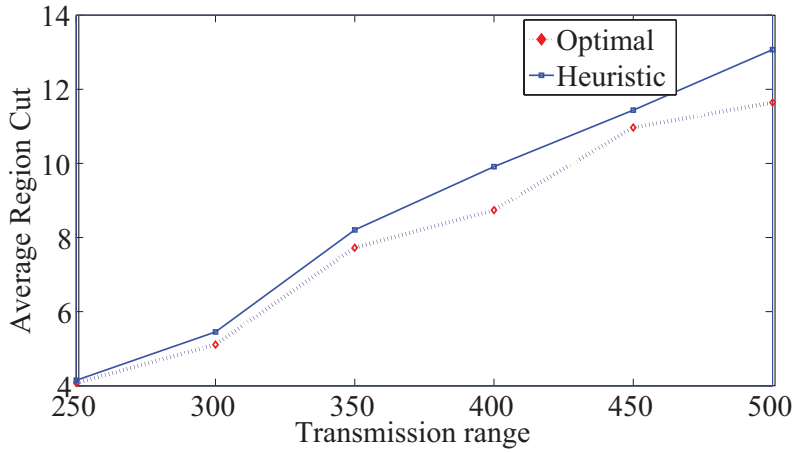


Figure 3.9: Average value of region Cut vs Transmission Range for a graph with 30 nodes and region radius of 100 m

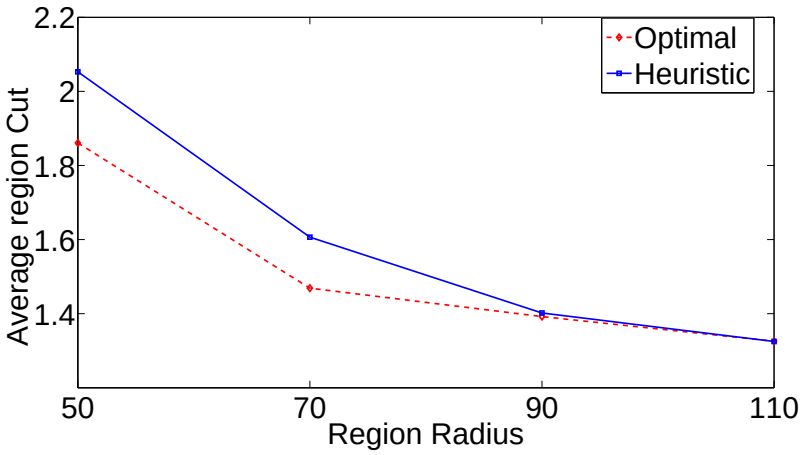


Figure 3.10: Average Region Cut vs Region Radius of a graph with 30 nodes and transmission range of 150 m

3.2.3.2 Results for the MRDP Problem

The efficacy of the region-disjoint paths heuristic is also evaluated through simulations. In this experiment, the average number of cases for which the heuristic solution differs from the optimal solution is measured. The results for this set of experiment are given in Figures 3.11 and 3.12. Figure 3.12 shows the percentage of cases where the number of solutions produced by heuristic differed from the optimal number of region-disjoint paths by i for $0 \leq i \leq 4$ and $i > 4$. It can be seen that in more than 87% of the cases, the number of region-disjoint paths produced by the heuristic differed from the optimal by at most 1. Figure 3.11 shows the variation of number of region-disjoint paths (averaged over all source-destination pairs) with the transmission range. It can be seen from the figure that increase in transmission range increases the number of region-disjoint paths. This is because the increase in transmission range induces more edges in the graph and thus more region-disjoint paths may be available.

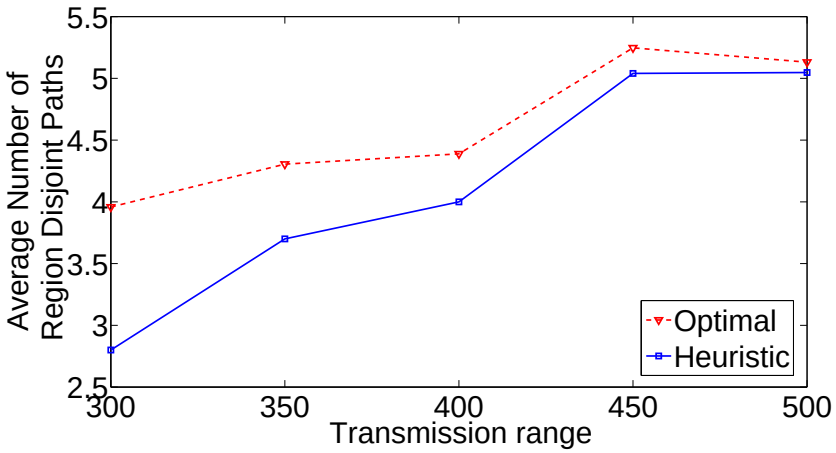


Figure 3.11: Average Number of Disjoint Paths vs Transmission Range in a graph of 30 nodes and region radius of 100 m

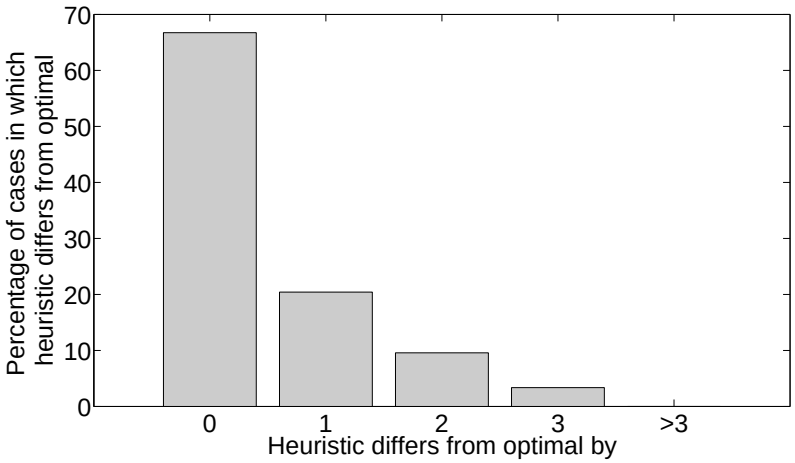


Figure 3.12: Percentage of cases in which number of region-disjoint paths computed by heuristic differs from optimal value for random graphs of size 30

3.3 Impact of Region-based Faults on the Connectivity of Wireless Sensor Network in Single Region Fault

Connectivity is an important parameter to measure the robustness of wireless sensor networks. Two nodes in a wireless sensor networks form a communication link if they are within signal transmission range of each other. Communication link between two nodes is a local phenomenon and depends on the various factors like transmission signal strength, antenna orientation, surrounding environment, obstructions etc. Two most widely accepted wireless communication models in literature are (i) *unit-disk graph model* and (ii) *log-normal shadow fading model*. In literature an important research interest is to find out the probability of connectivity of a network, under these two communication model, formed by a set of nodes distributed uniformly and randomly on a two dimensional plane. The series of works [5–7, 9, 10, 43–48] established many important and interesting results for an ad-hoc network formed in this manner. If $\kappa(G)$ and $d_{min}(G)$ denote the connectivity of the graph G formed by the distribution of nodes in DA and the minimum node degree of G, respectively, they provided techniques for finding answers to the questions of the following form,

- How to compute $P(d_{min}(G) \geq 1)$?
- How to compute $P(\kappa(G) \geq 1)$?
- Is $P(d_{min}(G) \geq 1)$ = $P(\kappa(G) \geq 1)$?
- What should the minimum transmission range r_t of the nodes be so that $\kappa(G) \geq 1$?
- What should the minimum transmission range r_t of the nodes be so that $\kappa(G) \geq 1$ with specified probability p?

where $P(d_{min}(G) \geq 1)$ and $P(\kappa(G) \geq 1)$ denote the probability of no isolated nodes in the graph G and probability of connectivity at least 1 respectively.

However, all these studies are either restricted to fault free network scenario where all the nodes are operational or assume that the faults are random in nature, *i.e.*, the probability of a node or link failing is independent of its location in the deployment area. This research extends the analysis of wireless network connectivity along the same line but in presence of a single region fault. Connectivity analysis in such localized fault scenario is considerably different than random fault or fault-free scenario. A set of distributed nodes in a deployment area, the network formed by the nodes and a fault region is shown in Figure 3.1. In this section the impact of *region-based faults* on the the connectivity of wireless networks, is studied, in (i) *unit-disk graph model*, (ii) *log-normal shadow fading model*. The research contributions on this topic are listed as follows:

1. Providing an analytical expression for computing $P(d_{min}(G) \geq 1)$ where node distribution is uniform and faults are region-based for both the communication model,

2. Through extensive simulation we find $P(\kappa(G) \geq 1)$ and compare it with $P(d_{min}(G) \geq 1)$ and

3. Study the impact of region based faults on the connectivity of the wireless network under unit disk graph and log-normal shadow fading model.

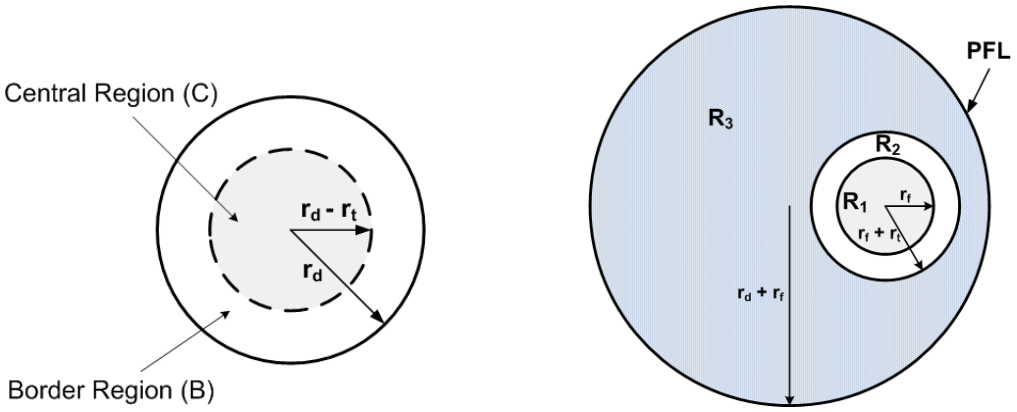
3.3.1 System Models

This section discusses the node distribution model, wireless channel model and fault model considered for this problem.

Node distribution: Nodes are assumed to be distributed randomly and uni-

formly on an infinitely large two dimensional plane with a constant node density ρ. A circular subarea $\mathbf{A}$ of radius r_d containing n nodes is considered from this infinite plane such that $\rho = n/A$. In limiting case for large n, the uniform distribution of n nodes in an area $\mathbf{A}$ is well approximated by Poisson point process.

Definition 6. Deployment Area (DA): *The area where the transceivers are deployed, assumed to be a circle of radius* r_d.



(a) Central(C) and Boundary(B) regions

(b) Regions of location of fault centers (R_1, R_2, R_3)

Figure 3.13: Sub-regions of DA and PFL

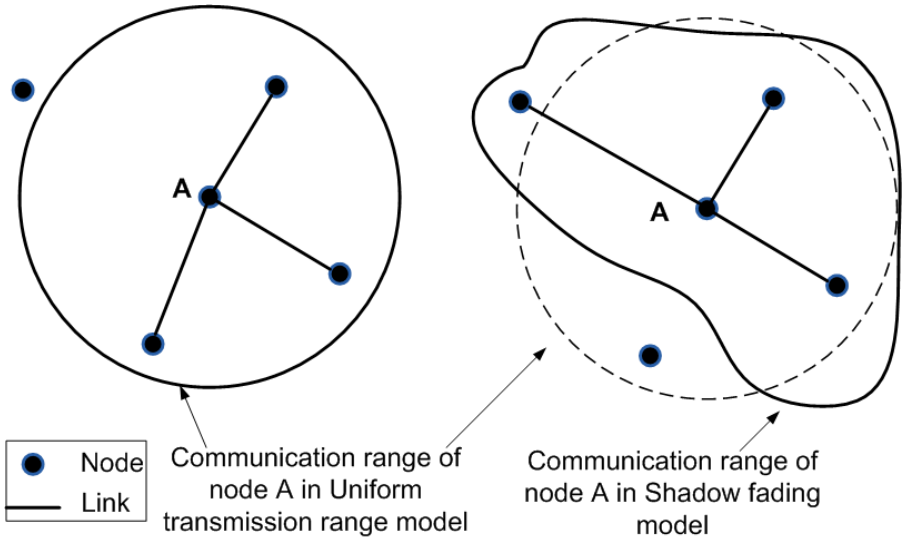


Figure 3.14: Links between nodes with and without shadowing effect

Signal propagation model

Two types of signal propagation models are considered:

Unit-Disk Graph Model (UDG): In wireless radio communication, received signal power decreases logarithimically as the distance between transmitter and receiver increases. This phenomenon is called path loss. If transmitter u transmits a signal at power $P_t(u)$ and receiver v receives the signal at power $P_r(v)$, then

$$\frac{P_r(v)}{P_t(u)} \propto \frac{1}{d^\alpha}$$

where d is the separation distance between transmitter and receiver and α is the path loss exponent. α depends on the propagation environment and terrain structure and can vary from 2 in free space to 6 in dense urban environment [49].

A wireless link is established between u and v if the received power $P_r(v)$ at v is greater than a threshold power $P_{th}(v)$. The most commonly used signal propagation model in wireless ad-hoc networks literature assumes that maximum transmission range of a transmitter is equal to the distance R_0 for which $P_r(v) = P_{th}(v)$ and the received signal power is uniform at any receiver v at a distance r, $0 \leq r \leq R_0$ [10]. So the probability of having a link between two nodes u and v at a distance r is a normalized function of distance $\hat{r} = r/R_0$ and is given by $p(\hat{r}) = 1$ if $\hat{r} \leq 1$ and $p(\hat{r}) = 0$ otherwise. The graph formed by this model is called *unit-disk* graph. In UDG model, r_t is considered to be the transmission radius of any node.

Definition 7. Transmission Coverage Area (TCA): *In UDG model TCA is the area covered by a transmitter,* i.e.*, if a receiver is in this area it will be able to decode the signal transmitted by the transmitter.* In this thesis it is assumed that the TCA is a circle of radius r_t and is the same for all the transmitters. In fault

scenario, there will be a partial or total loss of the coverage area of a transmitter if the TCA of the transmitter overlaps with a fault-region (FR).

Definition 8. Central Region (C): *The circular sub-region of DA with center being the center of the DA, and radius* $(r_d - r_t)$ *(Figure 3.13a).*

Definition 9. Boundary Region (B): *The annular sub-region of DA, being outside the central region C. It is concentric with the DA, with inner radius* $(r_d - r_t)$ *and outer radius* r_d *(Figure 3.13a).*

Log-normal Shadow fading Model(LSF): Wireless network graph constructed based on the above model may be highly inaccurate as it does not consider the variation of the signal strength due to presence of different obstructions in its path. Measurement studies of wireless signal power show that the mean power of signal at different distances from the transmitter varies log-normally (with standard deviation σ) around the area mean power [49]. The value of σ varies from 0 to as large as $12dB$ [49]. This model is known as *log-normal shadowing* model and is very close to real radio propagation model. The link probability between two nodes at distance r (as shown in [47] [48]) is given by :

$$p(r) = \frac{1}{2} - \frac{1}{2}\text{erf}\left(\frac{10\alpha}{\sqrt{2}\sigma}\log_{10}\frac{r}{R_0}dB\right). \tag{3.1}$$

where the normalization factor R_0 is the maximum distance at which a link can be formed in absence of shadowing effect , *i.e.*, at $\sigma = 0$ and erf(.) denotes the error function. The implication of this link probability is that some nodes at a normalized distance $\hat{r} < 1$ from the transmitter may not have a link while some nodes at a normalized distance $\hat{r} > 1$ may form a link with transmitter. Here it is assumed that all transmitters transmit at equal power, *i.e.*, R_0 is the same for all transmitters and all links formed in the network are undirected. Also it is

assumed that, given a specific value of α and σ, a node will have no neighbors beyond distance R if $p(r > R) \approx 0$. Then the area πR^2 around a node is called *neighbor-hood area* of the node.

Fault Model: In this chapter faults are considered to be localized and confined in a single region. Fault region is considered to be circular of radius of r_f. It is assumed that all the nodes in this fault region are faulty and do not participate in the network connectivity. Since the connectivity of the nodes is tested only within a circular subarea $\mathbf{A}$ of radius r_d, any region fault center occurring within a distance of $(r_d + r_f + R)$ from the center of $\mathbf{A}$, will either remove some of the nodes in area $\mathbf{A}$ or from the *neighbor-hood* of the nodes in area $\mathbf{A}$. It will be called Potential Fault Location area (*PFL*) in this thesis. Fault centers occurring outside this area will have no affect on the nodes of $\mathbf{A}$ or on their neighbors.

Definition 10. Fault Region (FR): *The area where the transceivers are out of operation due to faults. In this section it is assumed that the FR is a circle of radius* r_f*.*

Definition 11. Potential Fault Location area (PFL): *This is defined to be the area where the center of the faults may be located. PFL is defined as circle concentric to the DA, and having radius* $(r_d + r_f + R)$*. Fault centers outside this area will have no effect on the nodes in the DA.*

3.3.2 Probability Of Having No Isolated Nodes In The Graph

Let X and Y be the random variables denoting the location of a node in the area $\mathbf{A}$, and the location of the center of a Fault region (FR), respectively. In the node distribution model nodes are considered to be uniformly distributed over $\mathbf{A}$ with area $A = \pi r_d^2$. For these uniformly distributed nodes over finite area $\mathbf{A}$,

the probability density function (pdf) is given by:

$$f_X(\mathbf{x}) = \begin{cases} \frac{1}{A} & , \text{if} \quad \mathbf{x} \in \mathbf{A} \\ 0 & , \text{otherwise} \end{cases} \tag{3.2}$$

The node density of the area $\mathbf{A}$ is $\rho = n/A$. The distribution of Y is uniform over PFL with area $A_f = \pi(r_d + r_f + R)^2$ and probability density function (pdf):

$$h_Y(\mathbf{y}) = \begin{cases} \frac{1}{A_f} & , \text{if} \quad \mathbf{y} \in PFL, \\ 0 & , \text{otherwise} \end{cases} \tag{3.3}$$

In presence of a random fault region (FR), a node within a circle of πr_f^2 from the fault center will be considered *non-operational* or *dead*. Any node outside FR will be considered *operational* or *live*. Only live nodes will be considered to be in the network system. Let Z be another random variable which denotes location of a live node in the system after fault. Note that random variable Z can only take a subset of values that random variable X can take. So the probability density function of Z over $\mathbf{A}$ is a sub-probability density function [50] and consequently the probability of Z over $\mathbf{A}$ will not sum up to 1. Given a fault at location $\mathbf{y}$, region of overlap of FR with $\mathbf{A}$ is denoted by $\mathbf{A}'(\mathbf{y})$. The conditional sub-probability density function of $Z \mid Y$ is given by:

$$g_{Z|Y}(\mathbf{z} \mid \mathbf{y}) = \begin{cases} \frac{1}{A} & , \text{if} \quad \mathbf{z} \in \mathbf{A} \setminus \mathbf{A}'(\mathbf{y}) \\ 0 & , \text{otherwise} \end{cases} \tag{3.4}$$

Then sub-probability density function of Z is given by

$$\begin{aligned} g_Z(\mathbf{z}) &= \int\int_{A_f} g_{ZY}(\mathbf{z},\mathbf{y})\mathrm{d}\mathbf{y} = \int\int_{A_f} g_{Z|Y}(\mathbf{z} \mid \mathbf{y})h(\mathbf{y})\mathrm{d}\mathbf{y} \\ &= \frac{1}{AA_f}\int\int_{\substack{\|\mathbf{y}\| \le r_d + r_f + R \\ \|\mathbf{z}-\mathbf{y}\| > r_f}} \mathrm{d}\mathbf{y} \\ &= \frac{(A_f - \pi r_f^2)}{AA_f} = \bar{p} \quad , \text{if } \mathbf{z} \in A \\ &= 0 \qquad , \text{otherwise} \end{aligned} \tag{3.5}$$

where $\|.\|$ denotes the Euclidean distance of a location from the center of $\mathbf{A}$.

3.3.2.1 Probability of having no isolated nodes in the graph

Let us consider a node at location $\mathbf{x}$. The probability of node at $\mathbf{x}$ having a link with another node at a distance r is given in equation(3.1). Degree D of a node is the number of neighbors of that node. Let the expected degree of the node at $\mathbf{x}$ is $E(D \mid \mathbf{x}) = \mu(\mathbf{x})$. For Poisson point process the probability of a node having degree d is

$$P(D = d|\mathbf{x}) \approx \frac{\mu(\mathbf{x})^d}{d!} e^{-\mu(\mathbf{x})} \tag{3.6}$$

Then the probability that the given node is isolated is

$$P(\text{node iso}|\mathbf{x}) = P(D = 0|\mathbf{x}) \approx e^{-\mu(\mathbf{x})} \tag{3.7}$$

Thus the probability of having any isolated node is given by

$$P(\text{node iso}) = \int\int_A P(\text{node iso}|\mathbf{x}) g_Z(\mathbf{x})\mathrm{d}\mathbf{x} \tag{3.8}$$

Probability that none of the n nodes in subarea A are isolated is

$$P(\text{no node iso}) = (1 - P(\text{node iso}))^n \tag{3.9}$$

Again applying Poisson approximation,

$$P(\text{no node iso}) \approx \exp\Big(- n \int \int_A e^{-\mu(\mathbf{x})} g_Z(\mathbf{x}) \mathrm{d}\mathbf{x} \Big) \tag{3.10}$$

Next section gives the analytical expression, for calculating of the value of expected degree at any possible node location $\mathbf{x} \in A$, is given.

3.3.2.2 Expected degree of a node in A due to any random region fault

In absence of a fault, the expected value of D of a node X located at $\mathbf{x}$, can be computed by integrating $\rho p(r)$ over the entire system plane and is given as [47]:

$$E_1 = \rho \int_0^{2\pi} \int_0^{\infty} p(r) r \, dr \, d\theta \tag{3.11}$$

In presence of a fault with center Y located at $\mathbf{y}$, the expected degree of node X will decrease. Let $d(\mathbf{x}, \mathbf{y})$ be the distance between X and fault center Y. Then the loss of neighbors by node X at a distance $d(\mathbf{x}, \mathbf{y}) \geq r_f$ from fault center Y, without considering border effect, can be approximately given as a function of distance $d(\mathbf{x}, \mathbf{y})$

$$E_2(d(\mathbf{x}, \mathbf{y})) = \rho \frac{r_f}{4d(\mathbf{x}, \mathbf{y})} \int_0^{2\pi} \int_{d(\mathbf{x},\mathbf{y})-r_f}^{d(\mathbf{x},\mathbf{y})+r_f} p(r) r dr d\theta \tag{3.12}$$

The above expression is obtained by considering that the effect of fault will be symmetrical around the node in the annular area of width $2r_f$ at a distance between $d(\mathbf{x}, \mathbf{y}) - r_f$ and $d(\mathbf{x}, \mathbf{y}) + r_f$ from the node X (see Figure 3.15). So the net loss of neighbors due to a single fault at distance $d(\mathbf{x}, \mathbf{y})$ will be proportional to the value of the net loss of neighbors in this whole annular area. The multiplicative factor $\frac{r_f}{4d(\mathbf{x},\mathbf{y})}$ is the ratio of area of a single fault to the area of the whole annular region. Then expected degree of a node given the node location and fault location can be stated as

$$E(D|\mathbf{x}, \mathbf{y}) = \begin{cases} E_1 - E_2(d(\mathbf{x}, \mathbf{y})) & \text{, if } d(\mathbf{x}, \mathbf{y}) \geq r_f \\ 0 & \text{, if } d(\mathbf{x}, \mathbf{y}) < r_f \end{cases} \tag{3.13}$$

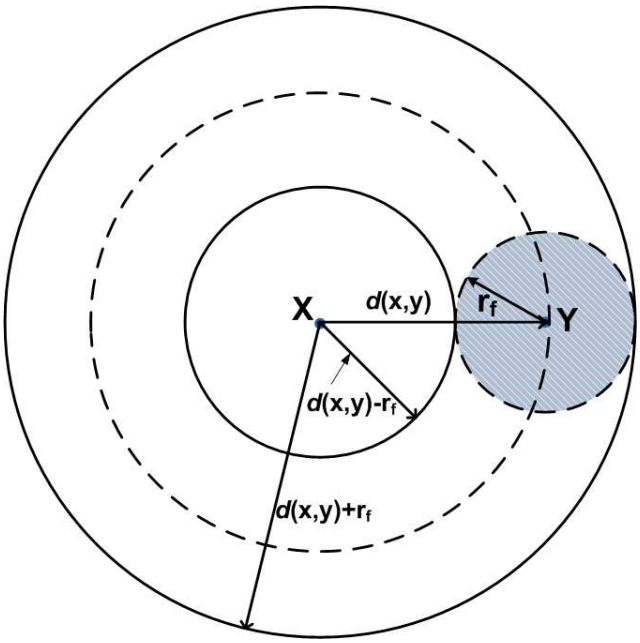


Figure 3.15: Effect of fault is symmetrical around the node at X

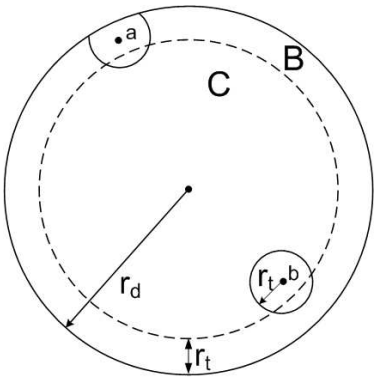
Expected degree of a node for any location of fault center in PFL can be given as

$$E(D|\mathbf{x}) = \int\int_{A_f} E(D|X,Y)h(\mathbf{y})\mathrm{d}\mathbf{y} \tag{3.14}$$

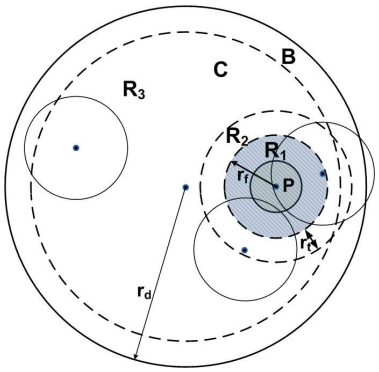
Using (3.14) in expression (3.10) we get the probability of no isolated node in the graph.

X and Y being random variables and independent of each other, Monte-Carlo integration [51] can be employed to numerically evaluate $P(\text{no node iso})$ in equation (3.10). N random points are generated for variable X uniformly in area $\mathbf{A}$ and for each X N random fault center are generated for variable Y uniformly within the fault region A_f. Then according to Monte-Carlo integration method if N is very large :

$$E(D|\mathbf{x_i}) = \frac{1}{N}\sum_{i}^{N} E(D|\mathbf{x_i},\mathbf{y_i})$$



(a) Circular deployment area without any region fault



(b) Circular deployment area with a region fault

Figure 3.16: Circular Deployment Areas with and without fault

and approximate value of $P(\text{no node iso})$ is given by:

$$P(\text{no node iso}) = \exp\Big(- n\bar{p}\frac{\pi r_d^2}{N}\sum_{i}^{N}\exp(-E(D|\mathbf{x_i}))\Big)$$

where $\mathbf{x_i}, \mathbf{y_i}$ are N points $i = 1, \ldots, N$ such that $\mathbf{x_i} \in \mathbf{A}$ and $\mathbf{y_i} \in A_f$. The evaluation is started with $N = 1000$ and increase N thereafter until equation (3.10) converges.

3.3.2.3 Expected degree of a node in DA due to any random region fault in UDG model and considering border effect

In *Unit-disk graph* model, since the transmission signal is constant within a radius r_t of a node, the expected degree of a node in DA can be easily calculated by computing the expected TCA of a node at **x** in the DA without any region fault and after a random region fault, respectively.

Area covered by a node in DA without any region fault:

Area covered by all nodes inside DA is not πr_t^2. Nodes in region B has coverage area less than πr_t^2 (node a in Figure 3.16a). This coverage area loss is called as *border loss*. Therefore expected value of area covered by a node

in DA without any region faults is $< \pi r_t^2$. The center of the DA is considered as the center of the co-ordinate system. Therefore any node at a distance u from the center of the DA can be represented with polar co-ordinates (u, ϕ), where $0 \leq \phi \leq 2\pi$. The coverage area of a node depends on magnitude of u, *i.e.*,

$$A_0(u) = \begin{cases} \pi r_t^2 & \text{for } 0 < u \leq r_d - r_t \\ A_0'(u) & \text{for } r_d - r_t < u \leq r_d \end{cases} \tag{3.15}$$

where $A_0'(u)$ is the intersection of two circles - one with center $(0, 0)$ and radius r_d, and another with center (u, ϕ) and radius r_t and is given by [52]

$$\begin{aligned} A_0'(u) &= r_t^2 \cos^{-1}\left(\frac{u^2 + r_t^2 - r_d^2}{2ur_t}\right) + r_d^2 \cos^{-1}\left(\frac{u^2 + r_d^2 - r_t^2}{2ur_d}\right) \\ &- \frac{1}{2}\sqrt{\left((r_d + r_t)^2 - u^2\right)\left(u^2 - (r_d - r_t)^2\right)} \end{aligned} \tag{3.16}$$

Area covered by a node in DA after any fault

It is evident that some nodes will suffer loss of coverage area for a region fault inside the DA (Figure 3.16b). Consider a node p inside DA. It will suffer total loss of coverage area of πr_t^2 due to any fault that has center within the region R_1, whereas it will suffer only partial loss of coverage area due to any fault with center within region R_2. Node p will not suffer any coverage area loss due to any fault located in $R_3 = PFL \setminus (R_1 \cup R_2)$. It is to be noted that a node can only be within the DA with area $A = \pi r_d^2$, whereas center of FR can be anywhere in PFL with area $A_f = \pi(r_d + r_f)^2$. In Table 3.1 all possible combination of the locations of the transceivers and location of the center of FR are shown. The expected coverage area for a node for each of these six cases can be calculated. An average of these coverage area will give the expected coverage area of a node due to a *region-fault*. Next a summary of the expressions for expected value of the area covered by a node in DA for all six cases is given.

Case	Transceiver Location	Fault Center Location
Case I	$C\ (0 \leq u \leq r_d - r_t)$	$R_1\ (0 \leq v \leq r_f)$
Case II	$C\ (0 \leq u \leq r_d - r_t)$	$R_2\ (r_f < v \leq r_f + r_t)$
Case III	$C\ (0 \leq u \leq r_d - r_t)$	$R_3\ (v > r_f + r_t)$
Case IV	$B\ (r_d - r_t < u \leq r_d)$	$R_1\ (0 \leq v \leq r_f)$
Case V	$B\ (r_d - r_t < u \leq r_d)$	$R_2\ (r_f < v \leq r_f + r_t)$
Case VI	$B\ (r_d - r_t < u \leq r_d)$	$R_3\ (v > r_f + r_t)$

Table 3.1: Locations of Transceiver and Relative Fault Center

In the following, the terms A_I, A_{II}, A'_{II}, A_{III}, A_{IV}, $A_V(u)$, $A_{VI}(u)$, all have dimensions $(area \times area)$, *i.e.*, $area^2$, since all of them represent some area integrated over another range of area. On the other hand, the terms $A_0(u)$, $A'_0(u)$, $A'_{II}(v)$, $G(u)$, A_1, A_2 all have dimensions of $area$.

Case I: If the center of FR is in region R_1, there is a complete loss of coverage area of the node after the fault. Therefore, integrating the area covered by the node over all possible positions of the center of fault in R_1 (Figure 3.17):

$$A_I = 0 \tag{3.17}$$

Case II: In case of the node being in region C and the location of center of the fault being in region R_2, FR will have partial overlap with the TCA of the node. In order to calculate this overlapping area, first the origin is shifted from the center of DA to the location of the node p. Without loss of generality, it can be considered that the polar co-ordinate of p as $(0, 0)$ after this shift. Since, the distance from p to the location of the center of FR is v, the polar co-ordinate of the center of FR can be similarly considered as $(v, 0)$. The overlapping area in this case is the area of intersection of a circle with center at $(v, 0)$ and radius r_f, and a circle with center at $(0, 0)$ and radius r_t (see Figure 3.18). Therefore, the coverage area loss of the node due to this particular fault is the intersection of

the two circles [52], and given by:

$$A'_{II}(v) = r_t^2 \cos^{-1}\left(\frac{v^2 + r_t^2 - r_f^2}{2vr_t}\right) + r_f^2 \cos^{-1}\left(\frac{v^2 + r_f^2 - r_t^2}{2vr_f}\right) - \frac{1}{2}\zeta$$

$$\text{where} \quad \zeta = \sqrt{\big((r_f + r_t)^2 - v^2\big)\big(v^2 - (r_f - r_t)^2\big)} \tag{3.18}$$

Integrating $A'_{II}(v)$ over the annular region R_2 (*i.e.*, varying v from r_f to $r_f + r_t$, and varying the polar co-ordinate angle from 0 to 2π)

$$\begin{aligned}
A'_{II} &= \int_{r_f}^{r_f + r_t} 2\pi v A'_{II}(v)\mathrm{d}v \\
&= 2\pi\Bigg[-\frac{r_t^2 r_f^2}{2}\tan^{-1}\left(\frac{r_t^2 + r_f^2 - v^2}{\zeta}\right) + \frac{v^2}{2}\Bigg(r_t^2 \cos^{-1}\left(\frac{r_t^2 - r_f^2 + v^2}{2r_t v}\right) \times \\
&+ r_f^2 \cos^{-1}\left(\frac{r_f^2 - r_t^2 + v^2}{2r_f v}\right)\Bigg) - \frac{\big(r_t^2 + r_f^2 + v^2\big)\zeta}{8}\Bigg]_{r_f}^{r_f + r_t} \\
&= \Bigg[\frac{\pi}{2}\Bigg(4r_f^2(r_f^2 - r_t^2)\sin^{-1}\left(\frac{r_t}{2r_f}\right) + 2\pi r_f^2 r_t^2 \\
&- r_t r_f (r_t^2 + 2r_f^2)\sqrt{1 - \frac{r_t^2}{4r_f^2}}\Bigg)\Bigg]
\end{aligned} \tag{3.19}$$

Therefore the integration of the area covered by the node for all possible locations of the center of faults in R_2 is given by

$$A_{II} = \pi r_t^2\Big(\pi(r_f + r_t)^2 - \pi r_f^2\Big) - A'_{II} \tag{3.20}$$

Case III: In this case, the node will not suffer any loss of coverage area due to region fault with fault center in R_3. The area covered by a node in C with transmission range r_t is πr_t^2, and integrating this coverage area over the range of v (*i.e.*, $r_f + r_t < v \leq r_d + r_f$) of all such faults with center at R_3

$$A_{III} = \pi r_t^2\Big(\pi\big((r_d + r_f)^2 - (r_f + r_t)^2\big)\Big) \tag{3.21}$$

Considering the above three cases, the expected value of the coverage area for the nodes in C after faults with fault centers in any of the regions R_1, R_2 or R_3,

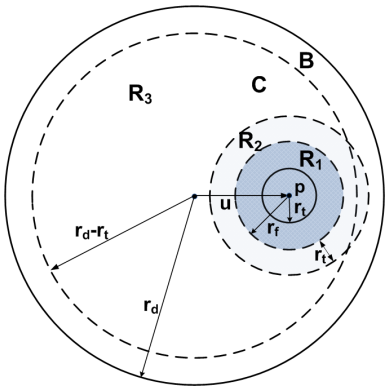


Figure 3.17: Node at point p is at a distance u from the center of DA and $0 \leq u \leq r_d - (r_t + r_f)$

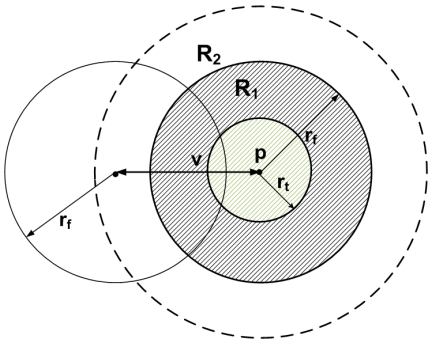


Figure 3.18: CASE II: Node at p inside C and fault center in R_2, *i.e.*, $0 \leq u \leq r_d - r_t$ and $r_f < v \leq r_f + r_t$

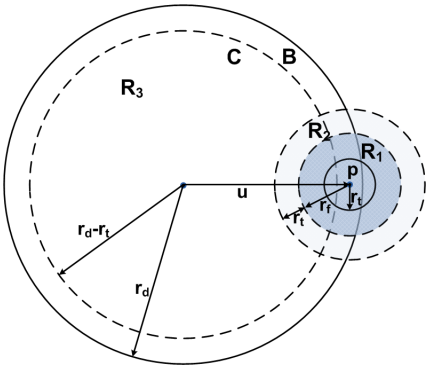


Figure 3.19: Node at point p is at a distance u from the center of DA and $r_d - r_t \leq x \leq r_d$

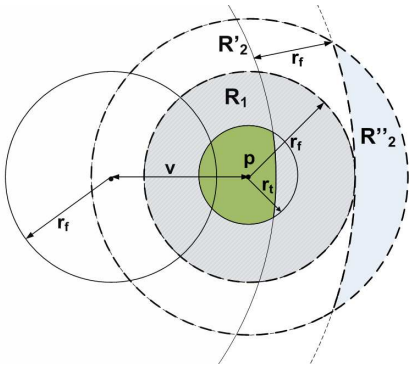


Figure 3.20: CASE V: Node at p inside B and fault center in R_2, *i.e.*, $r_d - r_t < u \leq r_d$ and $r_f < v \leq r_f + r_t$

can be computed as:

$$
\begin{aligned}
A_1 &= \frac{(A_I + A_{II} + A_{III})}{\pi(r_d + r_f)^2} \\
&= \frac{1}{\pi(r_d + r_f)^2}\left[\pi^2 r_t^2\Big((r_d + r_f)^2 - r_f^2\Big) - \frac{\pi}{2}\left(4r_f^2(r_f^2 - r_t^2)\sin^{-1}\left(\frac{r_t}{2r_f}\right)\right.\right. \\
&\quad \left.\left. +2\pi r_f^2 r_t^2 - r_t r_f (r_t^2 + 2r_f^2)\sqrt{1 - \frac{r_t^2}{4r_f^2}}\right)\right] \qquad (3.22)
\end{aligned}
$$

Since A_1 is independent of u (the distance of the node from the center of DA),

the expected degree of any node in the region C can be computed as

$$\mu_1 = \rho A_1 \tag{3.23}$$

Case IV: Similar to **C**ase I, since the center of FR is in region R_1, the entire area covered by the node is lost for the fault. Therefore, integrating over all such fault locations in the region R_1, we get (Figure 3.19):

$$A_{IV} = 0 \tag{3.24}$$

Case V: When the node is in the boundary region B and the relative location of the center of FR is in R_2, some part of the region R_2 will always be outside the potential location of fault center PFL. Let us define this location as R_2'' and $R_2' = R_2 \setminus R_2''$ (see Figure 3.20). In the absence of such a region R_2'' (*i.e.*, when $R_2'' = \phi$), the total area lost by the node for all such faults with centers in R_2 will be identical as **C**ase II, *i.e.*, A_{II}'. Therefore coverage area lost of the node within unit area of R_2 for such faults is

$$\frac{A_{II}'}{\text{area of } R_2} = \frac{A_{II}'}{\pi\big((r_f + r_t)^2 - r_f^2\big)} \tag{3.25}$$

Also, the coverage area of such a node without any fault is given by $A_0'(u)$ (from equation 3.16). Therefore integrating the coverage area for such a node over all such locations of the fault center:

$$A_V(u) = \left(A_0'(u) - \frac{A_{II}'}{\text{area of } R_2} \right) \times \text{area of } R_2' \tag{3.26}$$

The area of the region R_2' needs to be calculated next. The area of the region $R_2' \cup R_1$ is the area of intersection of two circles, one with center at $(0, 0)$ and radius $(r_d + r_f)$, and the other with center at $(u, 0)$ and radius $(r_f + r_t)$ (coordinates

in polar form). The area of this region can be calculated as

$$\begin{aligned} G(u) &= \pi(r_t + r_f)^2 - (r_t + r_f)^2 \cos^{-1}\Big(\frac{(r_d + r_f)^2 - u^2 - (r_t + r_f)^2}{2u(r_t + r_f)}\Big) \\ &+ (r_d + r_f)^2 \cos^{-1}\Big(\frac{u^2 + (r_d + r_f)^2 - (r_t + r_f)^2}{2u(r_d + r_f)}\Big) \\ &- \frac{1}{2}\sqrt{\Big((r_t + r_d + 2r_f)^2 - u^2\Big)\Big(u^2 - (r_d - r_t)^2\Big)} \end{aligned}$$

Area of R_1 is πr_f^2. Therefore, area of R_2' for a node at distance u from the center of the DA is given by

$$\text{area of } R_2' = (\text{area of } R_1 \cup R_2' - \text{area of } R_1) = G(u) - \pi r_f^2$$

From the above computations, the integration of the expected value of coverage area of the nodes in B over all such fault center locations in R_2 can be computed as

$$A_V(u) = \left(A_0'(u) - \frac{A_{II}'}{\pi\big((r_f + r_t)^2 - r_f^2\big)}\right)\Big(G(u) - \pi r_f^2\Big) \tag{3.27}$$

Case VI: Similar to **C**ase III, in this case also the coverage area of the node does not have any overlap with FR. From equation 3.16, the coverage area of the node without fault scenario is given by $A_0'(u)$ (where u is the distance of the node from the center of DA). Now, integrating this coverage area after fault over all possible locations of the fault center in regions R_3,

$$\begin{aligned} A_{VI}(u) &= A_0'(u) \times \text{area of } R_3 \\ &= A_0'(u) \times (\text{area of PFL} - \text{area of } R_1 \cup R_2') \\ &= A_0'(u)\Big(\pi(r_d + r_f)^2 - G(u)\Big) \end{aligned} \tag{3.28}$$

Considering the above three cases, the expected value of the coverage area of a node in region B and at distance u from the center of DA after fault is given by:

$$A_2(u) = \frac{A_{IV} + A_V(u) + A_{VI}(u)}{\pi(r_d + r_f)^2} \tag{3.29}$$

Therefore expected degree of any node in the boundary region B (at a distance u from the center of DA) after fault can be computed as

$$\mu_2(u) = \rho A_2(u) \tag{3.30}$$

3.3.3 Simulation results and Discussion

Extensive simulations are performed in order to study the relation between the probability of having an isolated node and the probability of the graph being connected at the presence of a random fault using both UDG and log-normal shadow fading model as communication model. The impact of the radius of FR r_f on these probabilities is also investigated.

Simulation Results in Unit-Disk Graph Model:

First, n nodes are uniformly and randomly distributed on a circular deployment area of radius r_d. Then a circular fault of radius r_f is generated with its center placed randomly over the circular area within distance $(r_d + r_f)$ from the center of DA. All the nodes (say n'), which lie in FR are removed and graph $G(V, E)$ is formed with the remaining $(n - n')$ number of nodes. The edges are formed between nodes whenever the distance between two such nodes is less than transmission radius r_t. Then the graph $G(V, E)$ is checked whether this is connected or has any isolated nodes. The java library Jgrapht is used to construct the random graphs by this process. In all these experiments radius of the deployment area is taken as 1000 unit and location of n nodes of the graphs are uniformly generated in a layout of $\pi(1000)^2$ square-unit area. Three parameters are considered that impact the probability of connectivity: (i) the number of nodes n, (ii) the region radius r_d and (iii) the transmission range r_t of the nodes. For a particular r_f and r_t, the same experiment is repeated for k_1 number of different node locations and k_2 number of times for different fault

locations, and finally, the percentage of connected topologies and the percentage of topologies with no isolated nodes is computed. If $k_1 k_2$ is large enough, this experiment gives fairly good estimate of probability of connectivity $P(\text{con})$ and probability of no isolated nodes $P(\text{no iso node})$ in the graph $G(V, E)$. In these experiments value of ratio r_f/r_d is varied from 0.2 to 0.6 at steps of 0.05 and value of r_t/r_d is varied from 0.07 to 0.25 at steps of 0.01 with $k_1 k_2$ taken as $10,000$.

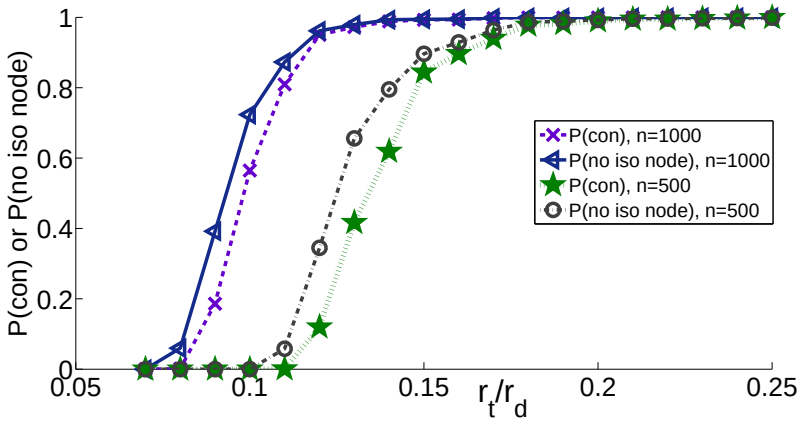


Figure 3.21: Probability $P(\text{no iso node})$ that the graph has no isolated node and comparison with $P(\text{con})$, when $r_f = 250$

In the first set of experiments, the probability of no isolated nodes in the graph is compared with probability of connectivity due to a random fault in the graph for different values of r_t/r_d. The value of r_t/r_d is varied from 0.07 to 0.25 for $n = 1000$ and 500. The same experiment is repeated for $r_f = 250$ unit and 400 unit. Results are shown in Figure 3.21 and Figure 3.22, respectively. It may be noted that for the same n probability of connectivity is always less than than probability of no isolated nodes in the graph for low value of r_t/r_d, but when r_t/r_d is relatively high both these probabilities merges. This means that even with a region fault, if nodes are distributed uniformly, probability of no isolated

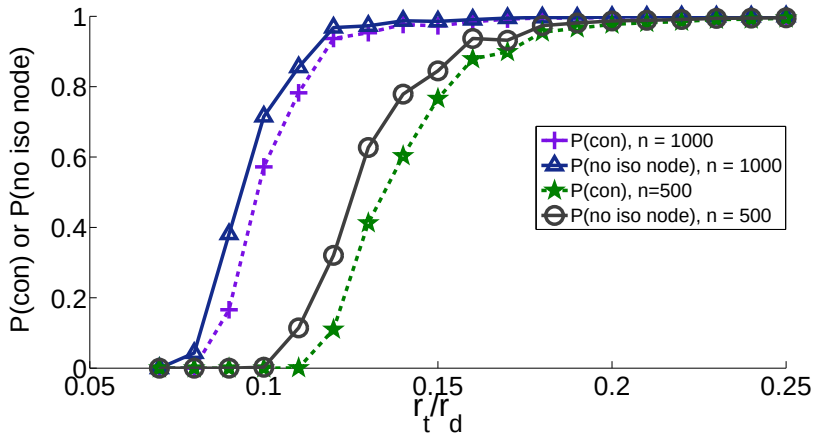


Figure 3.22: Probability $P(\text{no iso node})$ that the graph has no isolated node and comparison with $P(\text{con})$, when $r_f = 400$

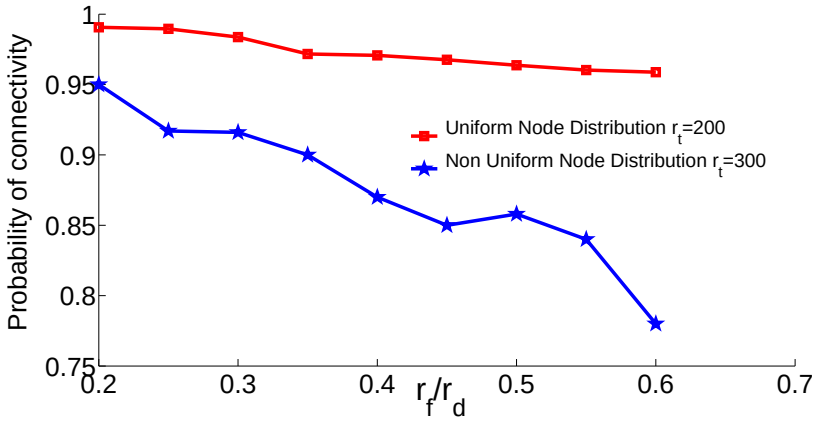


Figure 3.23: Effect of Fault Radius on the Probability of Connectivity for uniform and non-uniform distribution, number of nodes $(n) = 500$

nodes is a good estimate of probability of connectivity of the graph. Another noteworthy point is that the curve for $n = 1000$ is little steeper than the curve for $n = 500$, *i.e.*, probability of connectivity increases faster if the graph is dense.

In the second set of experiments, the effect of probability of connectivity in a graph is studied with increase of fault radius and the results are compared for graphs with two different node distribution - uniform and non-uniform (here, Gaussian). The value of r_f/r_d is varied from 0.2 to 0.6 for $n = 1000$

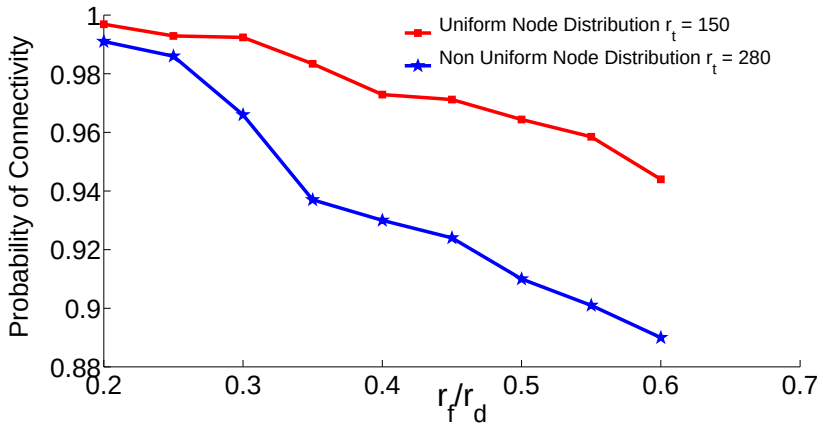


Figure 3.24: Effect of Fault Radius on the Probability of Connectivity for uniform and non-uniform distribution, number of nodes $(n) = 1000$

and $n = 500$. The transmission range r_t for each curve is fixed to the value which used to give connected graph with probability 1 without any fault. Results for graph with uniform and gaussian node distribution is shown in Figure 3.23 and Figure 3.24, respectively. An interesting observation is that probability of connectivity for uniform distribution doesn't decrease as sharply as probability of connectivity for non-uniform distribution. Also the decrease in probability of connectivity for uniform distribution is very small with the increase in fault radius r_f. This phenomenon can be explained using the following two choices of the experimental setup: (i) choice of circular DA and circular FR, (ii) uniformity of the node distribution in the DA.

Also the results of simulation is compared with the values obtained from the analytical result. For $r_f = 250$, the value of the ratio r_t/r_d is varied from 0.07 to 0.25 for $n = 1000$ and 500, and the value of $P(\text{no node iso})$ found from the simulations is compared with theoretical analysis. The results are shown in Figure 3.25. It can be seen that the simulations results follow closely to the values found from the theoretical analysis. Similar results are plotted for the

case $r_f = 400$ (Figure 3.26).

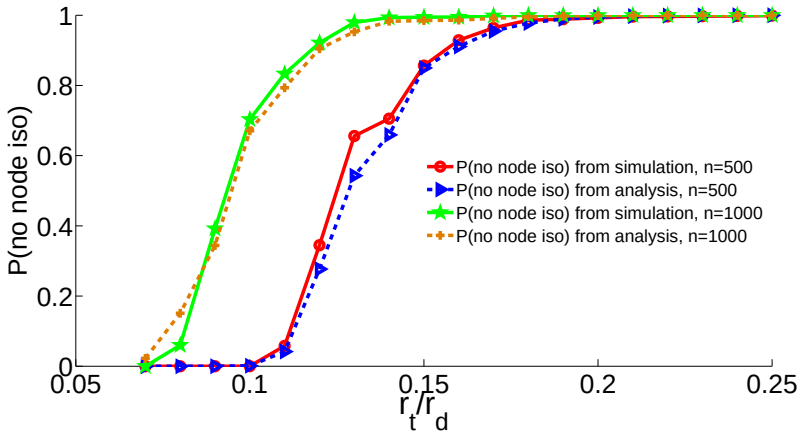


Figure 3.25: Comparison of $P(\text{no node iso})$ for simulations and analytical data for $r_f = 250$ and $n = 500$ and 1000

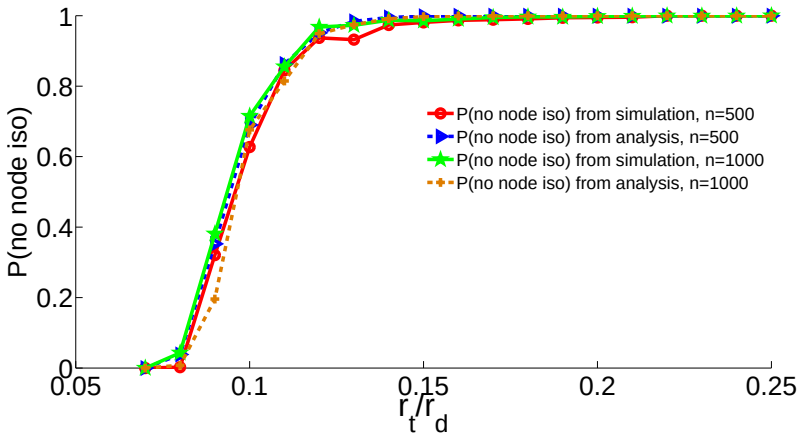


Figure 3.26: Comparison of $P(\text{no node iso})$ for simulations and analytical data for $r_f = 400$ and $n = 500$ and 1000

Simulation Results in Log-normal Shadow Fading Model:

In order to avoid border effect in log-normal shadow fading model, the simulation setting is changed a little bit. A circular simulation area $\mathbf{B}$ containing the circular observation subarea $\mathbf{DA}$ in the middle is considered. The probability of the graph being connected and the probability of no isolated nodes, only for the

nodes inside area DA are then tested. In order to minimize border effect of area $\mathbf{DA}$ we consider radius of $\mathbf{B}$ twice as the radius of $\mathbf{DA}$, *i.e.*, $2r_d$.

First, $n_B = \rho B$ nodes are uniformly and randomly distributed on the deployment area $\mathbf{B}$ to ensure the density of area $\mathbf{DA}$ remains ρ. Then a circular fault area of radius r_f is generated with its center placed randomly over the circular area within distance $(r_d + r_f + R)$ from the center of the deployment area $\mathbf{DA}$. All the nodes (say n'), which lie in fault region are removed and graph $G(V, E)$ is formed with the remaining $(n_B - n')$ nodes. The edges are formed between nodes following the link probability given in equation (3.1). Then it is checked whether area $\mathbf{DA}$ has any isolated nodes or there exists a path between any two pairs of nodes in $\mathbf{DA}$. In all these experiments area $\mathbf{DA}$ is taken as $2.5 \times 10^5\ m^2$. The impact of the two parameters on the probability of the graph being connected that are studied are: (i) the density of nodes n in $\mathbf{A}$ and (ii) the fault region radius r_f keeping other factors α, σ and R_0 (see equation (3.1)) as constant. For a particular r_f and ρ, the same experiment is repeated for k_1 number of different node locations and k_2 number of times for different fault locations, and finally, the percentage of connected topologies and the percentage of topologies with no isolated nodes are computed. If $k_1 k_2$ is large enough, this experiment gives fairly good estimate of probability of the graph being connected $P(\mathrm{conn})$ and probability of no isolated nodes $P(\text{no iso node})$ in area $\mathbf{DA}$. In these experiments, the value of r_f is varied from 0 to 80 at steps of 20 and the value of ρ is varied from 0.0001 to 0.0008 at steps of 0.00004 with $k_1 k_2$ taken as $10,000$. In all the experiments value of $\alpha = 3$, $\sigma = 4$, $R_0 = 43.34$ m and $R = 120$ m are considered.

In the first set of experiments, the probability of no isolated nodes in the graph is compared with probability of the graph being connected due to a region

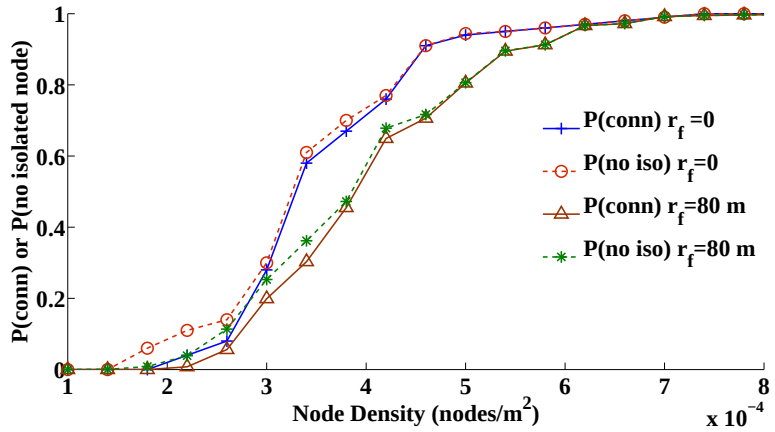


Figure 3.27: Probability that the graph has no isolated node $P(\text{no iso node})$ in comparison with Probability of Connectivity $P(\text{conn})$, when $r_f = 0$ and $r_f = 80$

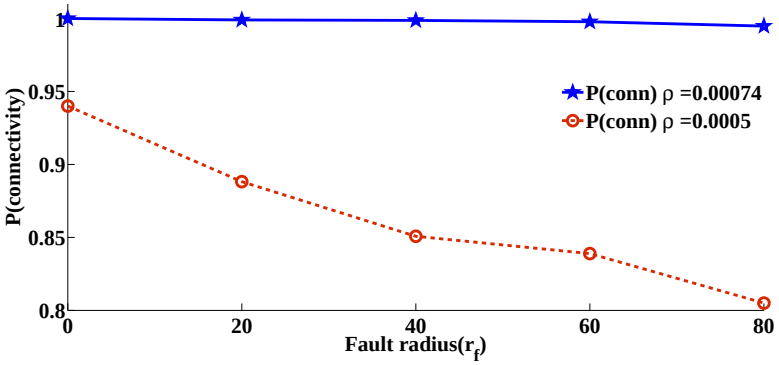


Figure 3.28: Effect of Fault Radius on the Probability of Connectivity $P(\text{conn})$, when $\rho = 0.00074$ and $\rho = 0.0005$ in Shadow fading model

based fault (in a random location) in the graph for different values of ρ. The value of ρ is varied from 0.0001 to 0.0008 for $r_f = 0$ and 80. Probability of the graph being connected decreases when fault radius is 80 ($r_f = 80$) as opposed to $r_f = 0$. Results are shown in Figure 3.27. It may be noted that for the same r_f probability of the graph being connected is always less than probability of no isolated nodes in the graph for low value of ρ, but when ρ is relatively high both these probabilities merges and eventually becomes 1. This means that even with region based faults, if nodes are distributed uniformly, probability of having

no isolated nodes in the graph is a good estimate of probability of connectivity of the graph.

In the second set of experiments, the impact of increase of fault radius on the probability of the graph being connected, is studied. The value of r_f is varied from 0 to 80 for $\rho = 0.00074$ and $\rho = 0.0005$. At node density $\rho = 0.00074$ the the graph is connected with probability 1 in fault free scenario (*i.e.*, $r_f = 0$). But with increase in fault radius the probability of connectivity decreases. Results for change of graph connectivity with fault radius is shown in Figure 3.28. An interesting observation is that decrease in probability of connectivity with the increase in fault radius r_f is insignificant when node density is high.

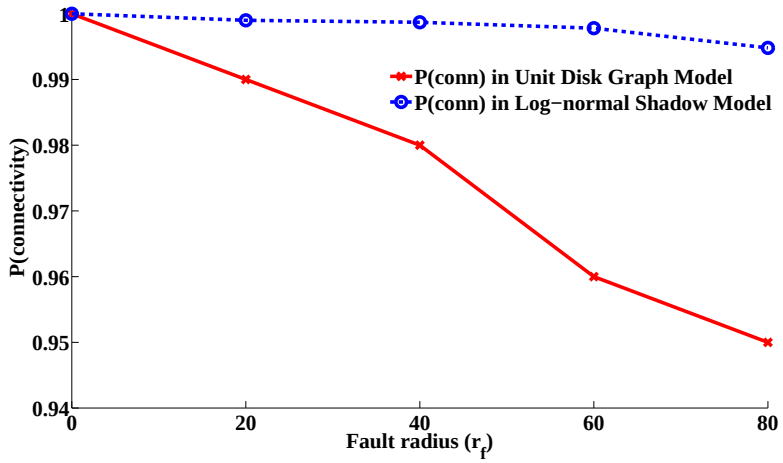


Figure 3.29: Effect of Fault Radius on the Probability of Connectivity $P(\text{conn})$, when $\rho = 0.00074$ for UDG and Shadow fading model

In third set of experiments, the effect of increase in fault radius on probability of the graph being connected (*P(conn)*), is studied in two different communication link models: i) Unit Disk graph (UDG) model and ii) Log-normal Shadow fading model. In order to make these two models comparable, we kept the expected degree of a node equal in no fault scenario for both these models. The

transmission range (r_t) of a node in UDG model is required to be set to 64 m to achieve the same. The value of r_f is varied from 0 to 80 keeping the node-density is $\rho = 0.00074$. Results for change of graph connectivity with fault radius is shown in Figure 3.29. As evident from the result, in both cases *P(conn)* decreases with increase in r_f, but for Log-normal Shadow model this decrease is significantly lower.

Chapter 4

DESIGN OF NETWORKS BASED ON REGION BASED COMPONENTS

The metric *region-based connectivity* incorporates the notion of *locality* of faults. However, both connectivity $\kappa(G)$ and region-based connectivity $\kappa_R(G)$ (where R is the region in which the faults are confined) of a graph G suffers from yet another limitation. These metrics provide information about the *network state* (*i.e.*, if the network is connected or not) as long as the number of faults do not exceed $\kappa(G)$ or $\kappa_R(G)$, respectively. *Neither $\kappa(G)$ or $\kappa_R(G)$ provide any information about the network state, if the number of faults exceeds these numbers.* The network state information that one may be interested in such a scenario are (i) the *number of connected components* into which G decomposes, (ii) the *size of the largest connected component*, (iii) the *size of the smallest connected component*.

These concepts are elaborated with the help of an example shown in Figure 4.1. It may be noted that the connectivity of both the *linear array* and the *star* networks is 1. Therefore no distinction between these networks can be made regarding their robustness (or fault-tolerance capability) using connectivity as the metric. However, the following observations can be made regarding the state of these two networks after failure of one node: (i) the linear array network can *break up into at most two components* and *the size of at least one component will be at least* $\lfloor n/2 \rfloor$, (ii) the star network can *break up into* $(n-1)$ *components* and *the size of these components can be as small as 1*, where n denotes the number of nodes in the network. In the unfortunate event of a network being disconnected after a failure, it is certainly desirable to have *a few, large connected*

components than *a large number of small connected components*. From operational point of view, a linear array network will certainly be preferable to a star network as it offers the possibility of a graceful performance degradation instead of a catastrophic failure. Unfortunately, the metric connectivity is incapable of making any distinction between these two networks.

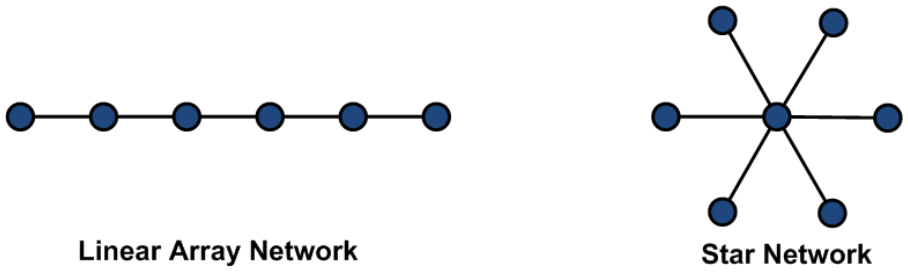


Figure 4.1: Linear Array Network and Star Network

In order to address this limitation, three new metrics for measuring the robustness (or fault tolerance capability) of a network, are introduced in this thesis. Suppose $\{R_1, \ldots, R_k\}$ is the set of all possible regions of graph G.

Definition 12. Region Based Component Decomposition Number (RBCDN): *Consider a k-dimensional vector C whose i-th entry, $C[i]$, indicates the number of connected components in which G decomposes when all nodes in R_i fail. Then, region-based component decomposition number (RBCDN) of graph G with region R is defined as $\alpha_R(G) = \max_{1 \leq i \leq k} C[i]$.*

Although, *RBCDN* measures the number of connected components, it does not capture the sizes of these components. For graceful degradation in performance, one may want that (i) there is at least one large component, or (ii) the size of even the smallest component is larger than a certain threshold value. In order to capture the size aspect of the disconnected components, two additional metrics are introduced.

Definition 13. Region Based Largest (Smallest) Component Size (RBLCS/RBSCS): *Consider a k-dimensional vector C_L (C_S) whose i-th entry, $C_L[i]$ $(C_S[i])$, indicates the size of the largest (smallest) connected component in which G decomposes when all nodes in R_i fail. Then, region-based largest component size (RBLCS) $\gamma_R(G)$ and region-based smallest component size (RBSCS) $\beta_R(G)$ of graph G with region R is defined as*

$$\gamma_R(G) = \min_{1 \leq i \leq k} C_L[i] \text{ and } \beta_R(G) = \min_{1 \leq i \leq k} C_S[i]$$

In order to have graceful degradation in performance, networks should be designed with a small value of $\alpha_R(G)$ and a high value of $\beta_R(G)$ and/or $\gamma_R(G)$. It may be noted that although the above metrics are defined for scenarios where faults are localized (*i.e.*, faults confined to a region), these concepts can easily be generalized for the scenario where *the faults are not localized.* The following two problems are studied in this research.

Problem 3. (Analysis): *Given the geometric layout of a graph $G = (V, E)$ on a 2-dimensional plane, and a region defined as a circular area of radius r, compute the* RBCDN, RBLCS *and* RBSCS *of G.*

Problem 4. (Design): *Suppose that the* RBCDN *and* RBLCS *of G with region R is $\alpha_R(G)$ and $\gamma_R(G)$ respectively. Suppose $\alpha_R(G)$ is considered to be too high for the application ($\gamma_R(G)$ is considered to be too low for the application) and it requires the* RBCDN *of the network not to exceed $\alpha_R(G) - K$ (the* RBLCS *of the network to be at least $\gamma_R(G) + K$), for some integer K. Assuming each additional link l_i that can be added to the network has a weight (cost) $w(i)$ associated with it, find the least cost link augmentation to the network so that its* RBCDN *is reduced from $\alpha_R(G)$ to $\alpha_R(G) - K$ (*RBLCS *is increased from $\gamma_R(G)$ to $\gamma_R(G) + K$).*

4.1 The RBCDN/RBSCS/RBLCS Algorithm

In this section, an algorithm is provided to computes the *RBCDN*, *RBSCS* and *RBLCS* of a graph $G = (V, E)$ when its layout in a plane is given as input and the region R is defined to be a circular area with radius r. The algorithm computes $\alpha_R(G), \beta_R(G)$ and $\gamma_R(G)$ in $O((n+m)^3)$ time, where $|V| = n$ and $|E| = m$. Specifically, the input to the algorithm are the following:

(i) a graph $G = (V, E)$ where $V = \{v_1, \ldots, v_n\}$ and $E = \{e_1, \ldots, e_m\}$ are the sets of nodes and links respectively, (ii) the layout of G on a 2-dimensional plane $LG = (P, L)$ where $P = \{p_1, \ldots, p_n\}$ and $L = \{l_1, \ldots, l_m\}$ are the sets of non-collinear points and straight lines on the two dimensional plane, (iii) a region is defined as a circular area R of radius r.

From the observations in Chapter 2, it is clear that one needs to examine only the Principal Regions (*i.e.*, the regions whose centers are at C-points) to compute the *RBCDN/RBSCS/RBLCS* of a graph G (with a layout LG on a 2-dimensional plane) and a circular region R of radius r. Since there are only $O((n+m)^2)$ of such regions, a polynomial time algorithm can be developed to compute RBCDN/RBSCS/RBLCS.

Algorithm 3 computes *RBCDN, RBSCS* and *RBLCS* of a network $G(V, E)$ with circular region R with radius r.

Theorem 3. *The complexity of the Algorithm 1 is* $O((n+m)^3)$.

Proof. As noted in Observation 5, time complexity of finding all the *C-points* is of $O((n+m)^2)$, where n and m is number of nodes and links in the network. So, time complexity of Step 1 is $O((n+m)^2)$. In Step 2, we have to check all

Algorithm 3: Computing RBCDN, RBSCS and RBLCS of a network graph $G = (V, E)$ with region R

Input :

1. The layout of graph $G = (V, E)$ on a two dimensional plane $LG = (P, L)$,
2. Region radius r,

Output: $\alpha_R(G)$, $\beta_R(G)$, $\gamma_R(G)$

1 Find the set of C-points using the algorithm sketched in Observation 5;
2 For each C-point c_j, find $G'_j = (V'_j, E'_j)$ a subgraph of G formed by removing the nodes and edges covered by region R_j centered at c_j;
3 For each such graph $G'_j = (V'_j, E'_j)$ find the largest connected component LC_j, smallest connected component SC_j and number of connected component CN_j using depth-first search [53]. Let LCS_j be the size of LC_j and SCS_j be the size of SC_j for graph $G'_j = (V'_j, E'_j)$;
4 $\alpha_R(G) = \max_j CN_j$, $\gamma_R(G) = \min_j LCS_j$ and $\beta_R(G) = \min_j SCS_j$;

edges in E and all nodes in V if they have intersections with the fault region R_j. So, time complexity of Step 2 is $O(n + m)$. Step 3 uses depth-first search to compute the number of connected components and has complexity of $O(\mid V \mid + \mid E \mid) = O(n+m)$. Step 2 and 3 is repeated $O((n+m)^2)$ of times. Therefore, the total time complexity of the Algorithm is $O((n + m)^3)$. □

4.2 Robust Network Design

The goal of robust network design problem is to have least cost augmentation of an existing network, so that it attains a specific target value of *RBCDN* and *RBLCS*. Formal description of the decision version of this problem is given below.

Problem 5 (*RBCDN* Reduction Problem (*RBCDN-RP*))**.** INSTANCE: *Given (i) a graph* $G = (V, E)$ *where* $V = \{v_1, \ldots, v_n\}$ *and* $E = \{e_1, \ldots, e_m\}$ *are the sets of nodes and links respectively,*

(ii) the layout of G *on a two dimensional plane* $LG = (P, L)$ *where* $P = \{p_1, \ldots, p_n\}$ *and* $L = \{l_1, \ldots, l_m\}$ *are the sets of points and lines on the 2-*

dimensional plane,

(iii) region R defined to be a circular area of radius r,

(iv) cost function $c(e) \in \mathbb{Z}^+, \forall e \in \bar{E}$, where $\bar{E}$ is complement of the link set E,

(v) integers C and K ($K \leq \alpha_R(G)$, where $\alpha_R(G)$ is the RBCDN *of G).*

QUESTION: *Is it possible to reduce the* RBCDN *of G by K by adding edges to G (from the set $\bar{E}$) so that the total cost of the added links is at most C?*

Problem 6 (*RBLCS* Augmentation Problem (*RBLCS-AP*))**.** INSTANCE: *Given*

(i) a graph $G = (V, E)$ where $V = \{v_1, \ldots, v_n\}$ and $E = \{e_1, \ldots, e_m\}$ are the sets of nodes and links respectively,

(ii) the layout of G on a two dimensional plane $LG = (P, L)$ where $P = \{p_1, \ldots, p_n\}$ and $L = \{l_1, \ldots, l_m\}$ are the sets of points and lines on the 2-dimensional plane,

(iii) region R defined to be a circular area of radius r,

*(iv) cost function $c(e) \in \mathbb{Z}^+, \forall e \in \bar{E}$, where $\bar{E}$ is complement of the link set E (*i.e.*, $\bar{E}$ is comprised of the links not present in E, but can be added to the graph $G = (V, E)$),*

(v) integers C and K ($K \leq n$).

QUESTION: *Is it possible to increase the* RBLCS *of G by K by adding edges to G (from the set $\bar{E}$) so that the total cost of the added links is at most C?*

It can be proved that both *RBCDN-RP* and *RBLCS-AP* are NP-complete by a transformation from the Hamiltonian Cycle in Planar Graph Problem (HCPGP) [42] which is known to be NP-complete. The proof can be found in Appendix A.

4.2.1 *Approximation Algorithm for the* RBCDN-RP *Problem*

This thesis presents an approximation algorithm for *RBCDN-RP* problem with an approximation factor of $O(\ln(K) + 2\ln(n+m)))$, where n and m are the number of nodes and links in the network graph $G = (V, E)$ and K is an integer by which $RBCDN\ \alpha_R(G)$ of G has to be reduced. Let G'_i be the subgraph induced from G by removing the links and the nodes intersecting with the region R_i and $CN(i)$ be the number of connected components in graph G'_i. For each region R_i if $CN(i) > \alpha_R(G) - K$ then at least $k_i = CN(i) - (\alpha_R(G) - K)$ links should be added to G'_i to reduce $CN(i)$ to $\alpha_R(G) - K$. Let l be the number of the regions where if fault strikes and the nodes of the region become inoperative, the graph decomposes to more than $\alpha_R(G) - K$ components.

As such only these l regions have to be considered for decreasing the $RBCDN$ of G from $\alpha_R(G)$ to $\alpha_R(G) - K$. Let $\mathcal{PE}_i$ be the set of the potential links that can be added between the connected components of G'_i to decrease its $CN(i)$. The potential links of G'_i are defined as the links not in E whose corresponding lines in the layout of the network do not have any intersection with region R_i. The links in $\mathcal{PE}_i$ are partitioned into disjoint subsets $\mathcal{PE}_{ij}, 1 \leq j \leq d(i)$, where each subset is non-empty and includes only the links between a pair of connected components of G'_i. The number $d(i)$ indicates the number of such disjoint subsets of $\mathcal{PE}_i$ for G'_i. The maximum value of $d(i)$ can be $\binom{CN(i)}{2}$.

In addition to the notations introduced in section 4.1 of this chapter, the following notations are used in the algorithms.

K: The integer by which $\alpha_R(G)$ of graph G with region R should be reduced.

k_i: The integer by which $CN(i)$ of G'_i should be reduced.

l: The number of regions R_i for which graph G'_i has $CN(i) > \alpha_R(G) - K$.

$\mathcal{PE}_i$: The set of the potential links that can be added between the connected components of G'_i to decrease its $CN(i)$.

$\mathcal{E} = \bigcup_{i=1}^{l} \mathcal{PE}_i$ and $m' = |\mathcal{S}|$; $\quad \mathcal{PE}_{ij}$: $\mathcal{PE}_{i,j} \subseteq \mathcal{PE}_i$.

It may be noted that more than one link $e \in \mathcal{PE}_{ij}$ can connect the same pair of components in G'_i.

$d(i)$: The number of subsets $\mathcal{PE}_{ij}$ present in $\mathcal{PE}_i$.

It should be noted $\bigcup_{j=1}^{d(i)} \mathcal{PE}_{ij} = \mathcal{PE}_i$ and $\forall j, k$, with $j \neq k,\ \mathcal{PE}_{ij} \cap \mathcal{PE}_{ik} = \emptyset$.

Algorithm 4 gives the approximation algorithm for *RBCDN-RP*. In each iteration, the algorithm chooses the most cost effective potential link. It may be noted that a potential link may appear in more than one $\mathcal{PE}_i$. The number of times a potential link e appears in different $\mathcal{PE}_i \in \mathcal{S}$ is called its *hit number* for that iteration and is denoted by H_e. The potential link whose cost ($c(e)$) to hit number (H_e) ratio is the smallest is considered to be the *most cost effective link*. If the link e_t, chosen in iteration t, has a hit in set $\mathcal{PE}_i$ then adding this link in graph G will reduce the $CN(i)$ of G'_i by 1. Therefore, k_i is decreased by 1. Adding two edges e_p and e_q from same subset $\mathcal{PE}_{ij} \subseteq \mathcal{PE}_i$ will not decrease the number of components of G'_i more than 1. As such, the subset $\mathcal{PE}_{ij}$ of $\mathcal{PE}_i$ which contains e_t is removed in iteration t. When G'_i attains its desired component number (*i.e.*, $k_i = 0$), it is not considered any more and $\mathcal{PE}_i$ is removed from $\mathcal{S}$.

Theorem 4. *The time complexity of Algorithm 4 is* $O(n^2(n+m)^2)$.

Proof. The inner for-$loop$ in Algorithm 4 runs for at most l times. In each iteration of outer $while$-$loop$, one link is selected and removed from all $\mathcal{PE}_i$. So outer $while$-$loop$ will only run at most m' times. So the complexity of Algorithm 4

Algorithm 4: Greedy Algorithm (GA) for *RBCDN-RP*

Input :

1. A set of all potential links $\mathcal{E} = \{e'_1, \ldots, e'_{m'}\}$,
2. a cost function $c(e)$ associated with link $e \in \mathcal{E}$,
3. $\mathcal{S} = \{\mathcal{PE}_1, \ldots, \mathcal{PE}_l\}$, $\mathcal{PE}_i \subseteq \mathcal{E}$,
4. a partition of each $\mathcal{PE}_i$ into $d(i)$ subsets, $1 \leq i \leq l$,
5. a number k_i associated with each $\mathcal{PE}_i$, $1 \leq i \leq l$.

Output: $\mathcal{E}' \subseteq \mathcal{E}$ such that $\sum_{e \in \mathcal{E}'} c(e)$ is minimum and $\mathcal{E}'$ hits at least k_i subsets of the partition of $\mathcal{PE}_i$

```
1 E' ← ∅;
2 while S ≠ ∅ do
3     Compute T = {H_1, ..., H_m'} such that ∀1 ≤ t ≤ m', H_t is the
      number of PE_i s, PE_i ∈ S that contains link e_t ∈ E ;
4     Pick e_t ∈ E such that c(e_t)/H_t is minimum;
5     E' ← E' ∪ e_t;
6     forall the PE_i such that e_t ∈ PE_i do
7         PE_i ← PE_i \ PE_ij, such that, e_t ∈ PE_ij;
8         k_i ← k_i − 1;
9         if k_i = 0 then S ← S \ PE_i;
```

will be $O(lm')$. Since in the worst case l is $O((n+m)^2)$ and m' is $O(n^2)$, the complexity of Algorithm 4 is $O(n^2(n+m)^2)$. □

Theorem 5. *Approximate solution produced by Algorithm 4 is at most* $O(\ln(K) + 2\ln(n+m)))$ *times the optimal solution.*

Proof. The subset $\mathcal{E}' \subseteq \mathcal{E}$ is chosen in such a way that $\mathcal{E}'$ hits at least k_i subsets in the partition of $\mathcal{PE}_i$ $\forall 1 \leq i \leq l$. So altogether $\lambda = \sum_{i=1}^{l} k_i$ subsets are hit by $\mathcal{E}'$. Let us order the subsets of each $\mathcal{PE}_i$, *i.e.*, $[\mathcal{PE}_{ij}, 1 \leq i \leq l, 1 \leq j \leq k_i]$ in the order in which they were hit by the elements of $\mathcal{E}'$ in the algorithm, resolving ties arbitrarily. Let the ordering be $\mathcal{S}' = \{\mathcal{PE}'_1, \mathcal{PE}'_2, \ldots, \mathcal{PE}'_\lambda\}$ where each $\mathcal{PE}'_i$ is some $\mathcal{PE}_{ij}, 1 \leq i \leq l, 1 \leq j \leq k_i$.

Assign a *price* for each subset $\mathcal{PE}'_i$, $\forall 1 \leq i \leq \lambda$, such that if $\mathcal{PE}'_i$ is hit by link e_t in some iteration the $price(\mathcal{PE}'_i) = \frac{c(e_t)}{H_t}$ where H_t is the number of subsets hit by e_t in that iteration. Let OPT be the total cost of the optimal solution. So at any iteration the leftover edges of the optimal solution set will hit the remaining sets of $\mathcal{S}'$ at a cost of at most OPT. In the worst case $\mathcal{PE}'_i$ will be hit by an edge in the i-th iteration. In i-th iteration at least $\lambda - i + 1$ elements remain in $\mathcal{S}'$. So the value of H_t in i-th iteration cannot be greater $\lambda - i + 1$. Since $\mathcal{PE}'_i$ will be hit by the most cost-effective link in this iteration, $price(\mathcal{PE}'_i) \leq \frac{OPT}{\lambda - i + 1}$. The cost of each link picked is distributed among the new subsets covered, the total cost of the link picked is $\sum_{e_t \in \mathcal{E}'} c(e_t) = \sum_{i=1}^{\lambda} price(\mathcal{PE}'_i) \leq \big(1 + \frac{1}{2} + \ldots + \frac{1}{\lambda}\big) OPT = O\big(\ln(\lambda)\big) OPT$. Also, $\lambda \leq K \times l$ and l is of the order of $O((n+m)^2)$. So the approximation ratio for Algorithm 4 is $O(\ln(K(n+m)^2))$. □

4.2.2 A Heuristic for RBLCS-AP *Problem*

For the RBLCS Problem the Heuristic given in Algorithm 5 is applied. G'_i be the subgraph induced from $G(V, E)$ by removing links and the nodes intersecting the region R_i and $\mathcal{C}_i$ be its set of connected components. Let similar to previous algorithm, $\mathcal{PE}_i$ be the set of the potential links that can be added between the connected components of G'_i to make a larger component. For each connected component p of G'_i let $size(p)$ be the number of nodes in the component, $AE(p)$ be the set of potential edges added in the component p and $cost(p)$ be a cost assigned to the component depending on the set $AE(p)$. Initially, $AE(p)$ is empty and $cost(p)$ is zero for all the primary components. Let us define LCS_i to be the size of the largest component of G'_i. Let l be the number of the regions in which if fault occurs, the graph G may decompose into components smaller than $\gamma_R(G) + K$. So only these l regions need to be considered to increase the $RBLCS$ of the graph G with region R from $\gamma_R(G)$ to $\gamma_{R(G)} + K$.

During each iteration of Algorithm 5, for all edges $e_i \in \mathcal{E}$, the $\varphi(e_i)$ is defined to be the total cost per unit size of the new components that can be constructed by adding e_i to all the graphs G'_j of regions R_j such that $e_i \in \mathcal{PE}_j$. In other words, $\varphi(e_i) = \frac{c(e_i)+\sum_{\{R_j:e_i\in\mathcal{PE}_j\}}(cost(p_j)+cost(q_j))}{\sum_{\{R_j:e_i\in\mathcal{PE}_j\}}(size(p_j)+size(q_j))}$ where p_j and q_j are the two components of G'_j that can be connected by edge e_i. Algorithm 6 describes how $\varphi(e_i)$ can be calculated for each edge e_i. Next, the algorithm selects the edge e_b which has the minimum φ among all other edges. It adds e_b to all G'_i such that $e_b \in \mathcal{PE}_i$ and update the value of LCS_i. If the updated $LCS_i \geq \gamma_{R(G)} + K$ then it does not consider the potential edges of this region in next iterations. For each newly created component p in the current iteration it sets $cost(p)$ as $size(p) \times \varphi(e_b)$. Before continuing to the next iteration it removes the edges in every remaining $\mathcal{PE}_i$ which connect the same pair of components as e_b. The algorithm continues until all regions has a component of size larger than $\gamma_{R(G)} + K$.

The following notations are used in the algorithms.

B: The desired largest component size for all regions, *i.e.*, $B = \gamma_R(G) + K$.

$LC(i)$: The largest component of the G'_i corresponding to the region R_i.

$LCS(i)$: The largest component size of the G'_i corresponding to the region R_i.

l: The number of regions R_i for which the graph G'_i has $LCS(i) < K$.

$\mathcal{PE}_i$: The set of the potential links that can be added between the connected components of G'_i. $\mathcal{E} = \bigcup_{i=1}^{l} \mathcal{PE}_i$.

$cp_i(e_j)$: The component resulted by adding the edge e_j connecting two components p and q of G'_i. Then, $cost(cp_i(e_j)) = (size(p) + size(q)) \times \varphi(e_j)$ and $AE(cp_i(e_j)) = AE(p) \cup AE(q) \cup \{e_j\}$.

It should be noted if even after adding all the edges in $\mathcal{E}$ to the graph G there exists some regions that do not have a component with size B then it is

Algorithm 5: Heuristic for *RBLCS-AP* Problem

Input : :

1. The desired RBLCS value (B),
2. A set of all potential edges $\mathcal{E} = \{e_1, \ldots, e_{m'}\}$,
3. a cost function $c(e)$ associated with edge $e \in \mathcal{E}$,
4. $\mathcal{S} = \{\mathcal{PE}_1, \ldots, \mathcal{PE}_l\}$, $\mathcal{PE}_i \subset \mathcal{E}$,
5. $\mathcal{C}_i$: The set of components of G'_i, $\forall i$.

Output: :

$\mathcal{E}' \subseteq \mathcal{E}$ such that $\sum_{e \in \mathcal{E}'} c(e)$ is minimum and G'_i has at least one component of size $\geq B$, $\forall 1 \leq i \leq l$.

$\mathcal{E}' \leftarrow \emptyset$;

while $\mathcal{S} \neq \emptyset$ *and* $\mathcal{E} \neq \emptyset$ **do**

  Find the edge $e_b \in \mathcal{E}$ with minimum φ using Algorithm 6;

  forall the $R_i, \forall 1 \leq i \leq l$ **do**

    if $e_b \in \mathcal{PE}_i$ *and connecting two components* p *and* q **then**

      Update $\mathcal{C}_i$ by replacing the two components p and q with $cp_i(e_b)$;

      Remove from $\mathcal{PE}_i$ all the other potential edges which connect the components p and q;

      if $size(cp_i(e_b)) > LCS_i$ **then**

        $LCS_i \leftarrow size(cp_i(e_b))$ and $LC_i \leftarrow cp_i(e_b)$;

      if $LCS_i \geq B$ **then**

        $\mathcal{E}' \leftarrow \mathcal{E}' \cup AE(LC_i)$;

        $\mathcal{S} \leftarrow \mathcal{S} - \{\mathcal{PE}_i\}$;

  $\mathcal{E} \leftarrow \mathcal{E} - \{e_b\}$;

return $\mathcal{E}'$;

infeasible to make RBLCS $= B$. In the worst case the heuristic will add all the edges in $\mathcal{E}$. So it will always find a solution to the $RBLCS$ problem if a feasible solution exists.

Algorithm 6: Finding $\varphi(e_i)$ for the edge e_i

```
1 Cost ← c(e_i), Weight ← 0;
2 forall the R_j, 1 ≤ j ≤ l do
3     if e_i ∈ PE_j and connecting the two components p and q then
4         Cost ← Cost + cost(p) + cost(q);
5         Weight ← Weight + size(p) + size(q);
6 φ(e_i) ← Cost / Weight;
```

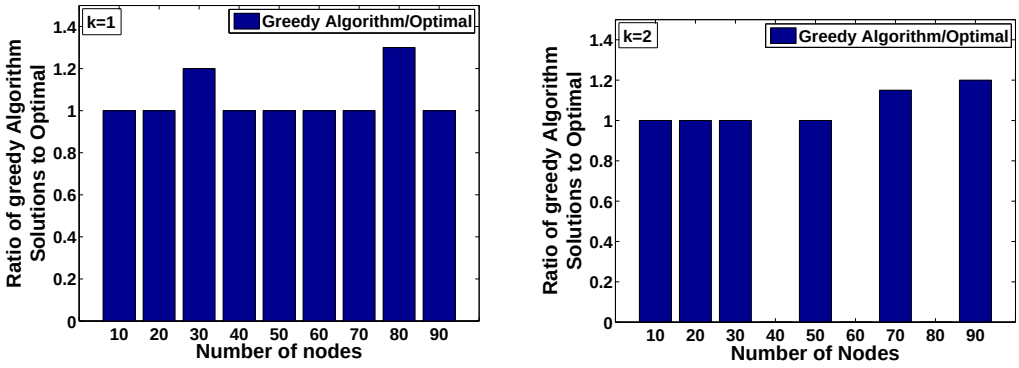
Theorem 6. *The time complexity of Algorithm 1 is* $O(n^4(n+m)^2)$.

Proof. In order to find $\varphi(e_i)$ for an edge e_i one will have to consider all the l distinct regions which is $O((n+m)^2)$. So, finding the edge with minimum φ (line 4) is $O(|\mathcal{E}| \times (n+m)^2) = O(m'(n+m)^2) = O(n^2(n+m)^2)$. In *RBLCS* problem $|\mathcal{E}| = O(n^2)$, the maximum number of links that can be added to the graph G. The for-$loop$ in line 5 is repeated l times. Also, the $while$-$loop$ in line 3 is repeated at most $|\mathcal{E}|$ times. Therefore, the complexity of the Algorithm 5 will be $O(n^2(n^2(n+m)^2)) = O(n^4(n+m)^2)$. □

4.3 Experimental Results for Accuracy of RBCDN and RBLCS Algorithm

This section presents 2 experimental results to demonstrate the efficacy of the proposed Algorithms 4 and 5 by comparing it's solution to the optimal results through extensive simulations. Optimal solution, for each experiment, are obtained by solving Integer Linear Program (ILP) using CPLEX Optimizer 10.0.

In both the experiments several random instances of network layout in a 2-dimensional plane are generated. In every instance the node locations are uniformly and randomly distributed on a square deployment area of side length 100 units. Erdos-Renyi model of random graphs is followed for this simulation. The probability of having a link between two nodes was chosen to be 0.3, so that the resulting graph do not become too dense or too sparse. For all the simula-



(a) $K = 1$ (b) $K = 2$

Figure 4.2: Comparison of the Solution of Algorithm 4 with the Optimal. The solutions of Algorithm 4 is compared by taking their ratios to the optimal. On the y-axis, ratio value 1 indicates that Algorithm 4 achieves optimal solution. Blank columns indicate the infeasibility of the corresponding instances.

tion experiments, the fault region was considered to be a circular area of radius 25 units. For every problem instance, first the set of C-points $\mathcal{C} = \{C_1, \ldots, C_T\}$ is computed and then using Algorithm 3, the $RBCDN$ and $RBLCS$ are computed. Also, $\forall C_i \in \mathcal{C}$, G'_i, $CN(i)$, $\mathcal{PE}_i$ and $d(i)$ (please refer to section 4.2.1 for the notations), are computed.

In first set of experiments the efficiency of the solution of approximation Algorithm 4 is compared for *RBCDN-RP* with the optimal solution for 10 random instances of network layout. The number of nodes, in these instances are varied from 10 to 90 in a step of 10. For each instance the algorithm is executed for values of K (number by which *RBCDN* needs to be reduced) as 1 and 2. For this case study, the results of the experiments are shown in Figure 4.2. The ratio of the cost of the solutions of the greedy algorithm (Algorithm 4) to the optimal cost for different values of n and K is plotted. It can be seen that in most of the cases, the ratio is 1, indicating that the greedy algorithm produces the optimal solution. In other cases the ratio is close to 1, indicating the greedy algorithm

produces a near optimal solution. In all of these cases the greedy algorithm takes a fraction of time required to find the optimal solution. It may be noted that for those instances of RBCDN-RP, where $\exists\ C_i \in \mathcal{C}$ such that $k_i > d(i)$, there exists no feasible solution. The instances with no feasible solutions can be seen as blank columns in the plots of Figure 4.2.

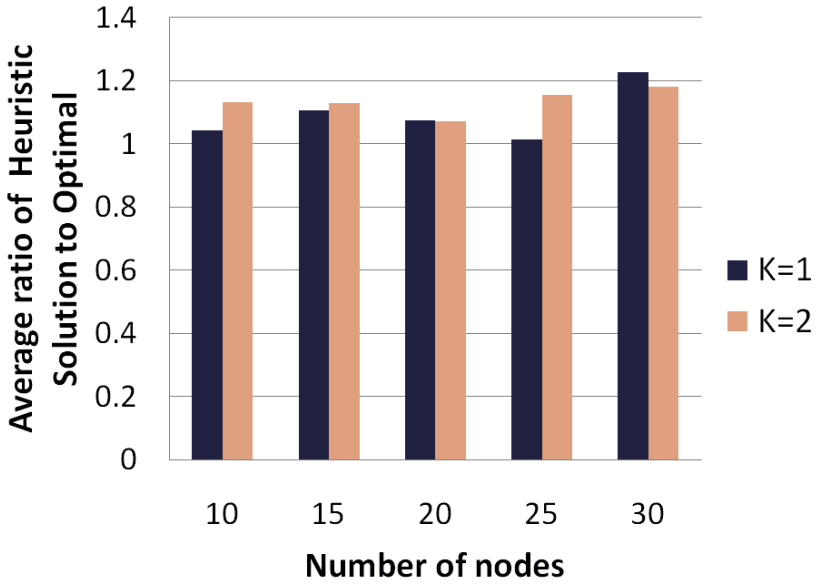


Figure 4.3: Comparison of the solution of the Heuristic (Algorithm 5) for *RBLCS-AP* with the Optimal by taking its ratio to the optimal.

In second set of experiments the results of the heuristic for *RBLCS-AP* are compared with the optimal solution through extensive simulations. The number of nodes, n, in these instances are varied from 10 to 30 in a step of 5. For each value of n, 10 random instances of network layout are created. The edge between every two nodes is added with probability $p = 0.5$. The heuristic finds a solution within a wide range of choices of p and n. However, when $p < 0.5$ and $n > 30$ the optimal algorithm takes an unacceptably long time (several days) to complete. This is due to the fact that when $p < 0.5$ and $n > 30$ the set of the potential edges becomes large. Since the complexity of the optimal algorithm is exponential, with a large set of potential edges, it takes an unacceptably

long computation time. Since the goal of the experiment was to compare performance of the heuristic with the optimal, the experiments are restricted to $p = 0.5$ and $n < 30$.

For each instance the algorithm is executed for values of K (number by which *RBLCS* needs to be augmented) equal to 1 and 2. For $n > 20$ and $p = 0.5$, almost all the instances do not have a feasible solution for values of $K \geq 3$. Therefore the value of K is restricted to be 1 and 2 only. The results of the experiments are shown in Figure 4.3. The average of the ratios of the cost of the solution of the heuristic to the optimal cost for each value of n and K is plotted. It can be observed that in all of these cases, the ratios are less than 1.4, indicating the superior quality of the heuristic solution. The heuristic executes in a small fraction of time needed to compute the optimal solution in all these experiments.

Chapter 5

DATA DISTRIBUTION SCHEME IN DATA STORAGE NETWORKS IN PRESENCE OF REGION-BASED FAULTS

Distributed storage of data files, shared among the users spanning over the entire network, reduces the file retrieval cost and improves fault tolerance. One of the simplest techniques of storing a data file F of size $|F|$ distributively over a network is to replicate the file l times and store each replica at different nodes in the network. Although this file replication scheme can tolerate failure of up to $l-1$ nodes, it has two major shortcomings, (i) the total storage space used by this scheme over the network is $l|F|$, and (ii) if only one node storing the file replica is compromised, an unauthorized user can have access to the entire file. In order to avoid these shortcomings, and enhance fault-tolerance, security and load balancing capability, error-correcting codes have been used extensively in data storage systems and server clusters - such as RAID [54] and DPSS [55]. One well known scheme for this purpose is the use of $(\mathcal{N},\mathcal{K})$ erasure codes [56–59]. In $(\mathcal{N},\mathcal{K})$ *maximum distance separable* (MDS) erasure code based storage system $\mathcal{N}$ coded segments of size $|F|/\mathcal{K}$ are created from the original file F and are stored in $\mathcal{N}$ nodes of the network (one coded segment per node). The advantage of this storage scheme is that the original file F can be reconstructed by any user in the network just by retrieving and then decoding any $\mathcal{K}$ out of $\mathcal{N}$ segments. Clearly location of the storage nodes within the network will have an impact on ease of retrieval of the segments.

The focus of this research is to design a robust file distribution scheme that takes into account the network topology - particularly in the scenario when one or more of the network nodes are unavailable due to failure.

This thesis considers the scenario where failure of nodes (storing data segments) might disconnect the network. $(\mathcal{N}, \mathcal{K})$ codes ensure that as long as $\mathcal{K}$ file segments survive the file can be reconstructed. However, this condition alone is not sufficient for successful file reconstruction where the network splits up into two or more connected components and although more than $\mathcal{K}$ segments survive, none of the components have more than $\mathcal{K} - 1$ segments. To be able to reconstruct the file, one has to ensure that at least one of the connected components has at least $\mathcal{K}$ segments. The file distribution scheme presented in this research utilizes $(\mathcal{N}, \mathcal{K})$ erasure codes and ensures that even in the event of a network disconnection due to node failures, one of the largest connected components will have at least $\mathcal{K}$ distinct file segments, with which to reconstruct the entire file. Since storage of file segments involves cost, the distribution scheme presented here also ensures that the total storage required over the network is minimized. In this model a node represents a user with some storage capacity. Rationale for ensuring that a largest connected component has sufficient number of segments to reconstruct the file, is that such a capability will benefit the largest number of users. Goal of the file distribution scheme is minimization of storage requirement. It is driven by the fact that although storage has become less expensive in recent times, cost of storing large data sets (of the order of petabytes, exabytes or higher) is still significantly high.

In a network spanning a large geographical area, the faulty nodes may be *spatially correlated* (*i.e.*, confined to a *region*). Such failures are often encountered in diaster situations, either natural (earthquake, forest fire, flood or

hurricane) or man-made (EMP attack or enemy bomb) causes, where only the nodes in the disaster zone are affected. These faults are generally termed as *spatially correlated faults* or *region-based faults*. A network spanning across a large area might get disconnected to several connected components due to such massive but localized failures. As a consequence, design of a data distribution scheme robust against such failures, is extremely important.

Contributions in this research are as follows:

- Provide a robust file distribution scheme utilizing $(\mathcal{N}, \mathcal{K})$ coding that ensures that even when the network is fractured into components due to region-based node and link failures, at least one of the largest component will have at least $\mathcal{K}$ segments to reconstruct the file. The scheme also minimizes the total data storage (σ) required over the network.
- Provide an Integer Linear Program for computing the optimal solution of data distribution problem (DDP).
- Provide a polynomial time approximation algorithm for the DDP on networks with an arbitrary topology, with a guaranteed performance bound of $O(\ln n)$.

5.1 Motivating Examples

Let us first demonstrate through an example how coding can help in reducing storage requirement for the robust data distribution scheme designed to tackle region-based faults. The term “robust” is used to imply that the distribution scheme enables the non-faulty nodes of a largest connected component to reconstruct the entire file after a region-based fault strikes the network.

The examples shown in Figure 5.1 (a) and (b) demonstrate the data distribution schemes with and without coding. Both the schemes are robust against region-based faults, where a region is defined to be a subgraph of diameter one. It is assumed that the storage capacity of each node is one. The Figure 5.1 shows distribution of three uncoded file segments A, B and C of a file F. As shown in Figure 5.1(a), the uncoded segments A, B and C must be stored in at least six nodes of the network, in order to make it robust against a region-based fault. However, as shown in Figure 5.1(b), if $(\mathcal{N}, \mathcal{K})$ coding is used and an extra coded segment $(A \bigoplus B \bigoplus C)$ is created ($\bigoplus$ represents an XOR operation), ($\mathcal{N} = 4$ and $\mathcal{K} = 3$ in this example), it is sufficient to store the data segments $A, B,\ C$ and $(A \bigotimes B \bigotimes C)$ in at most four nodes of the network, in order to make it robust against a same size region-based fault.

Figure 5.2 shows $(\mathcal{N}, \mathcal{K})$ coding based data distribution scheme on the European fiber backbone network from a major network service provider [60], in presence of a massive region-based fault. Fault is assumed to be a circular area of radius 150 miles. In this example, $\mathcal{N} = 20$ and $\mathcal{K} = 10$. The colored nodes in the Figure 5.2 are the nodes where a coded segment is stored. Nodes with same color store the same coded segment. Each node is assumed to have storage capacity of one. The distribution shown in Figure 5.2 stores data segments in ($\sigma = 22$) locations. In this example twenty-two is the fewest number of locations where the segments have to be stored to ensure robustness against any circular region-based fault of radius $r = 150$ miles, *i.e.*, the largest component of the network will have at least $\mathcal{K}$ nodes with distinctly coded file segments.

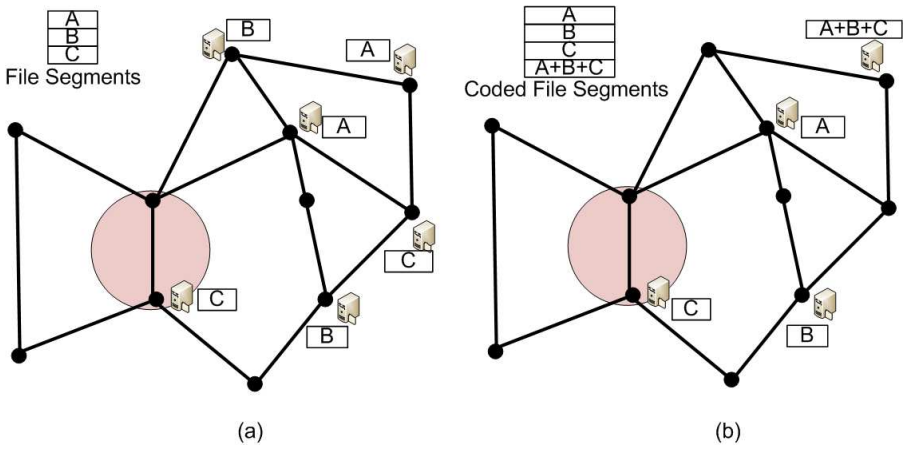


Figure 5.1: A data storage network with a region-based fault. Fig (a) uses no coding scheme. Storage used is 6 in this case. Fig (b) uses $(\mathcal{N}, \mathcal{K})$ coding scheme and uses only 4 storage units.

5.2 Related Work

Error-correcting codes have been used extensively in enhancing the performance, reliability and fault tolerance capability in data storage systems [56, 58, 59, 61–66]. In [56, 58], Dimakis *et al.*, consider node failures, and use erasure codes to solve the repair problem of node, (*i.e.*, how to replace the data on a failed node). However the problem under study in this research is considerably different from the one in [56, 58]. While in [56, 58] authors focus on the repair problem and study the trade-off between storage and bandwidth requirement, this study is directed towards development of a robust scheme that allows reconstruction of file after a few nodes disappear due to a region-based failure. Moreover, our technique accounts for the topology of the network while designing the data distribution scheme.

Distribution of coded file segments among different nodes in the network, taking topology of the network into account, has been studied in [64], [65], [66], [62], [57], [63]. The focus of this line of research is in developing a file distribu-

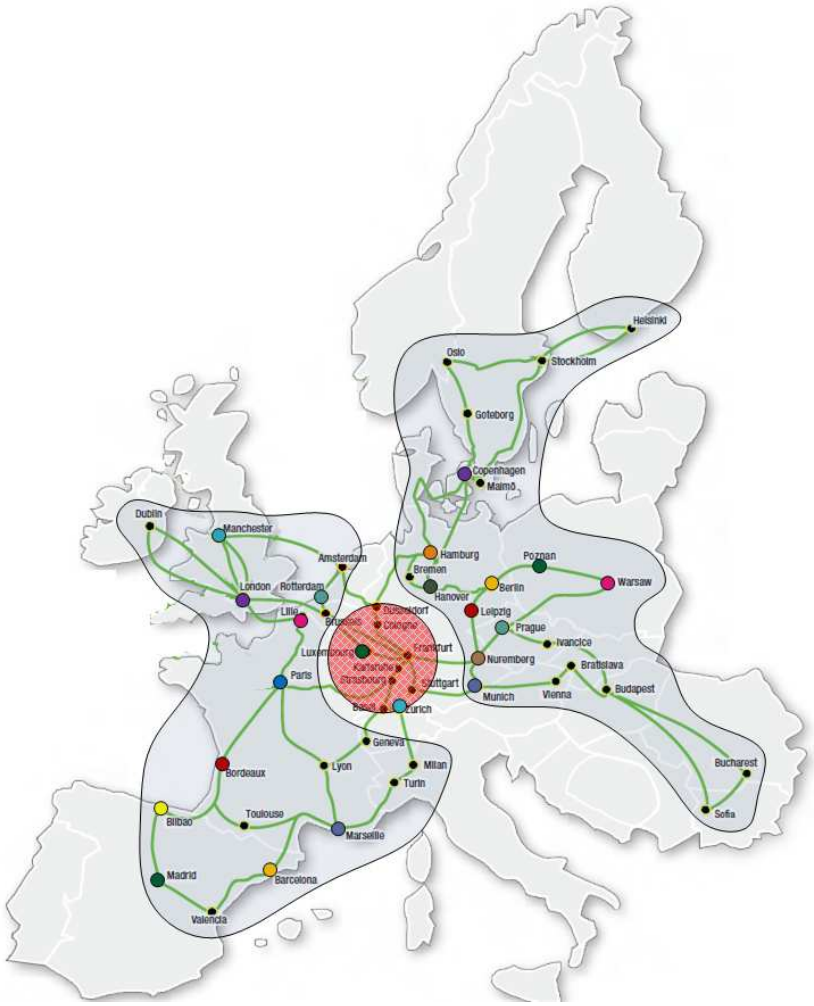


Figure 5.2: Fiber backbone of a major Europe network provider with a region-fault and distribution of file segments

tion scheme so that each node in the network can recreate the original file by accessing $\mathcal{K}$ file segments from nodes within their r hop neighborhoods. Jiang *et al.*, solves this problem optimally for networks with special structure such as trees [65] or torus [62]. This research is along the same line as that of [64] and [65] and our objective is also to minimize the total storage in the network. However, there exists a major distinction between this research and that of [64] and [65].

In [64–66], authors solve the problem when there is no fault in the network, whereas we discuss the file segment distribution scheme on network of arbitrary topology considering a region-based fault in the network.

In [67], Jiang *et al.*, solves the file segment distribution problem with the goal of minimizing $\mathcal{N}$ subject to the constraint that any node v in the network will have at least $\mathcal{K}$ distinct file segments within a specified radius r. However, minimizing $\mathcal{N}$ does not necessarily minimize the amount of storage (σ) that will be necessary to satisfy the proximity requirement. The example shown in Figure 5.3, shows that σ is 5 when the goal is to minimize $\mathcal{N}$ and σ is 4 when the goal is to minimize total storage (while satisfying the proximity requirement that the entire file be reconstructible by collecting $\mathcal{K}$ segments within one hop neighbors). In this example it is assumed each node can store at most one file segment. This example shows a tradeoff between $\mathcal{N}$ and σ. The storage requirement decreases with increase in $\mathcal{N}$. This thesis focuses on minimizing (σ) over the entire network.

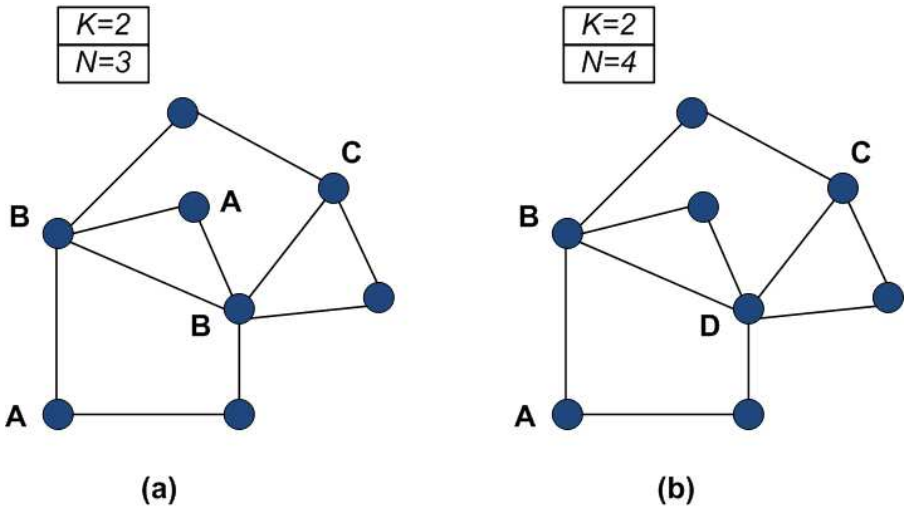


Figure 5.3: Example showing the tradeoff between the value of $\mathcal{N}$ and storage σ required in network. Total file segments available in Fig (a) are $\{A, B, C\}$, so storage required is 5 while file segments available in Fig (b) are $\{A, B, C, D\}$, so storage required is 4.

5.3 Data Distribution Problem Formulation

A file F of length $\mathcal{M}$ is split into $\mathcal{K}$ segments each of length $\mathcal{M}/\mathcal{K}$. Using erasure coding techniques like maximum distance separable (MDS) codes, Reed-Solomon (RS) codes, etc., these $\mathcal{K}$ segments can be encoded to get altogether $\mathcal{N}$ segments ($\mathcal{N} \geq \mathcal{K}$). According to the erasure coding theory, retrieving any $\mathcal{K}$ segments out of $\mathcal{N}$ segments is sufficient to reconstruct the whole file F.

The network is represented by a graph $G = (V, E)$ where V is the set of n nodes in the network and E is the set of links between them. Each node has a storage capacity for storing the file segments. For the file F, a node $v_i \in V$ can store maximum of α_i file segments. Nodes in the network are assumed to have both storage and processing unit. So any node can recreate the file F by downloading and decoding $\mathcal{K}$ appropriate coded segments from other nodes. Let the layout of G on a 2-dimensional plane be $LG = (P, L)$ where $P = \{p_1, \ldots, p_n\}$ and $L = \{l_1, \ldots, l_m\}$ are the sets of points (representing nodes) and straight lines (representing links) respectively. A region is considered to be a circular area R on this plane of radius r. Also it is assumed that due to a single region-fault all the nodes and links confined in the region are destroyed. Total number of such distinct regions that need to be considered for a graph layout LG on 2-dimensional plane is $t = O(n^2)$ for wireless networks [4] and $t = O((n+m)^2)$ for wired networks (refer Chapter 2). The difference in the number of distinct regions in wireless and wired network is because of the absence of any physical links in wireless network. Let $\mathcal{R} = \{R_1, R_2, \ldots, R_t\}$ be the set of distinct regions.

Due to a region-based fault the residual network might get disconnected. The data distribution scheme will try to distribute at most $\mathcal{N}$ encoded segments

of file F among the n storage locations of the network in such a way that the largest component of the residual network has at least $\mathcal{K}$ distinct segments to reconstruct the file. Let S_i be the set of nodes in a largest residual component in network G after region R_i fails, where $i = 1, \ldots, t$. Let set $\mathcal{S} = \{S_1, \ldots, S_t\}$ represents the set of t largest residual components of network G. Lets say after distribution of the coded segments, each node v_i receives ρ_i segments, such that $0 \leq \rho_i \leq \alpha_i$. Since storing each file segments involves a cost, the objective in this research is to minimize the total storage used in the whole network. Let $\mathcal{C} = \{1, 2, \ldots, \mathcal{N}\}$ be a set of $\mathcal{N}$ distinct colors. The problem can be formulated as color assignment problem where each color represents a segment and assigning a color $c_j,\ 1 \leq j \leq \mathcal{N}$ to a node v_i implies that file segment j is stored in node v_i. If the storage capacity of node v_i is α_i then at most α_i colors can be assigned to node v_i. In this chapter the terms color capacity and storage capacity of a node are used interchangeably.

Formally the data file segment distribution problem can be stated as:

Problem 7 (Data Distribution Problem (*DDP*))**.** INSTANCE: *Given*

(i) a graph $G = (V, E)$ *where* $V = \{v_1, \ldots, v_n\}$ *and* $E = \{e_1, \ldots, e_m\}$ *are the sets of nodes and links respectively,*

(ii) the layout of G *on a two dimensional plane* $LG = (P, L)$ *where* $P = \{p_1, \ldots, p_n\}$ *and* $L = \{l_1, \ldots, l_m\}$ *are the sets of points and lines on the 2-dimensional plane,*

(iii) maximum of α_i *colors can be assigned to a node* v_i*,*

(iv) region R *defined to be a circular area of radius* r*,*

(v) a set $\mathcal{S} = \{S_1, \ldots, S_t\}$ *where* $S_i \subseteq V$*,* $1 \leq i \leq t$*,*

(vi) a set of $C = \{1, 2, \ldots, \mathcal{N}\}$ *distinct colors and parameter* $\mathcal{K}$*.*

QUESTION*: Assign at most $\mathcal{N}$ colors to the nodes in V in such a way that (i) $\rho_i \leq \alpha_i$, where ρ_i is the number of colors assigned to node v_i, (ii) each set $S_i \in \mathcal{S}$ has at least $\mathcal{K}$ distinct colors and (iii) $\sigma = \sum_{i=1}^{n} \rho_i$ is minimum.*

Data distribution problem (DDP) can be shown to be NP-complete by transforming Hitting Set, a known NP-complete problem [42], to a special case of Data distribution problem (DDP).

5.4 Algorithms for Data Distribution Problems

In this section an optimal algorithm solution and a polynomial time approximation algorithm for the data distribution problem in any arbitrary network are presented.

5.4.1 Optimal solution for DDP in arbitrary networks

First, an integer linear program is formulated to solve the DDP optimally. For each node $v_i \in V$ and each color $c_j \in C$ a variable x_{ij} is used. x_{ij} is 1 if and only if the node v_i is given the color c_j. Similarly, for each set $S_p \in \mathcal{S}$ and each color $c_j \in C$ a variable z_{pj} is introduced. The variable z_{pj} is 1 if and only if the set S_p has at least one node assigned color c_j. Then constraint (1) means node v_i can be assigned at most α_i colors. Constraints (2) and (3) together mean that each set S_p should have at least $\mathcal{K}$ distinct colors. Then minimizing $\sum_{i}^{n} \sum_{j}^{\mathcal{N}} x_{ij}$ minimizes the total number of colors assigned to different nodes. In other words, the objective is minimizing the total storage space used all over the network. The ILP will solve the problem in exponential time.

Variables: For each node v_i and color c_j

$$x_{ij} = \begin{cases} 1, & \text{if node } v_i \text{ is assigned color } c_j \\ 0, & \text{otherwise.} \end{cases}$$

For each set S_p and color c_j

$$z_{pj} = \begin{cases} 1, & \text{if set } S_p \text{ has at least one node of color } c_j \\ 0, & \text{otherwise.} \end{cases}$$

$$\min \quad \sum_{i=1}^{n} \sum_{j=1}^{\mathcal{N}} x_{ij}$$

$$s.t. \quad \sum_{j=1}^{N} x_{ij} \leq \alpha_i, \quad \forall i = 1, \cdots, n \tag{5.1}$$

$$z_{pj} \leq \sum_{i:v_i \in S_p} x_{ij}, \quad \forall p = 1, \ldots, t, \ j = 1, \ldots, \mathcal{N} \tag{5.2}$$

$$\sum_{j=1}^{\mathcal{N}} z_{pj} \geq \mathcal{K}, \quad \forall p = 1, \ldots, t \tag{5.3}$$

$$x_{ij} \in \{0, 1\}, \ z_{pj} \in \{0, 1\}, \quad \forall x_{ij} \text{ and } z_{pj} \tag{5.4}$$

5.4.2 Approximation Solution for DDP in arbitrary networks

Next an approximation (Algorithm 7) (DDA) is presented to solve DDP in polynomial time for any arbitrary network. Step 1 and step 2 of DDA computes all the distinct regions of radius r on the graph layout LG and finds the largest component of the residual graph for each region fault. In Chapter 2, it has been shown that number of such distinct circular regions (t) in a geographical network is bounded by a polynomial function of number of nodes n and number of links m and can be computed in $O((n+m)^2)$. Set C_i maintains the list of

colors assigned to node v_i and α_i is the maximum color capacity of node v_i. k_j represents the number of distinct colors required for set S_j. Initially C_i, for all nodes $v_i \in V$, is empty and k_j is $\mathcal{K}$ for all sets $S_j \in \mathcal{S}$. The value of $\mathcal{N}$ is taken to be greater than total storage available in the largest component S_j, *i.e.*, $\mathcal{N} = \max_j \sum_{v_i \in S_j} \alpha_i$.

In each iteration, the *while-loop* from step (7-18) assigns a new color to a node. $\mathcal{T}$ is a set of integers H_i, where H_i denotes the number of times v_i appears in all the remaining sets in $\mathcal{S}$. A high value of H_i means v_i is a good candidate for assigning a new color. In each iteration, step 11 picks up the node v_p for which value of H_p is maximum among all the nodes in $\mathcal{T}$ with at least unit color capacity left. The *for-loop* from step (13-16) runs over all the sets S_j that contain v_p. Step 14 chooses the set of colors D_j which has not been assigned to any node in S_j. Step 17 chooses the color c which is minimum over all the colors $\bigcap_{S_j : v_p \in S_j} D_j$ selected within the *for-loop* and assigns c to the node v_p. Note that this intersection will be non-empty because total number of colors available is $\mathcal{N} = \max_j \sum_{v_i \in S_j} \alpha_i$. Also choosing the lowest color $c \in \bigcap_{S_j : v_p \in S_j} D_j$ ensures that all the sets S_j considered in the *for-loop* get a new distinct color. So the value of k_j is decreased in step 15 for all $S_j \ni v_p$. If in this process any S_j satisfies its all color requirement it is removed from $\mathcal{S}$. Step 18 reduces the maximum color capacity of node v_p by 1. If at any iteration $\alpha_p = 0, \forall v_p \in V$, the step 9 implies that all the nodes have been assigned colors up to its capacity. However, there are still some sets left $S_j \in \mathcal{S}$ which have not satisfied the coloring constraints of having $\mathcal{K}$ distinct colors. It implies no feasible solution is possible for that instance and the algorithm terminates. At the end, steps from (20-23) perform deletion of any redundant assigned colors. The double *for-loop* iterates through the list of the colors assigned to the nodes in reverse order and

deletes a color from a node if it turns out that removing the color from that node still produces a feasible solution.

Theorem 7. *The time complexity of DDA is* $O(n(n+m)^2)$.

Proof. In step 1 and 2 of Algorithm 7, the largest components of all the distinct regions for the layout LG of the graph G can be computed in $O((n+m)^2)$ time using the technique stated in Chapter 2. Number of distinct regions t in wired network is $O((n+m)^2)$. The inner *for-loop* satisfies the color requirement for at least one set S_j in worst case and reduces k_j by 1. The outer *while-loop* terminates when color requirements of all the sets are satisfied. So in worst case the outer *while-loop* will run for $t\mathcal{K}$ times which is $O((n+m)^2)$. The two *for-loops* in the reverse delete steps (20-23) will run for $\sum_i^n \rho_i$ times which is $O(n)$. For checking feasibility of the solution one will have to go through all the sets S_j in $\mathcal{S}$. So it will take $O(t)$ or $O((n+m)^2)$ time. So altogether the complexity of the algorithm is $O(n(n+m)^2)$. □

Theorem 8. *Approximate solution produced by DDA is at most* $O(\ln(K) + 2\ln(n+m))$ *times the optimal solution.*

Proof. The outer *while-loop* step (7-18) of algorithm DDA returns a set of nodes $A \subseteq V$ and a set of colors C_i assigned to every node $v_i \in A$. Since total $\mathcal{N}$ distinct colors available in color set $\mathcal{C}$ and $\mathcal{N}$ is greater than total color capacity $\sum_{v \in S_i} \alpha_v, \forall S_i \in \mathcal{S}$, in worst case all the colors assigned to the nodes in A will be distinct. Assigning a color to a node is equivalent to selecting a storage element from that node. So at the end of the *while-loop* the algorithm will return a set of storage elements $\mathcal{U} = \{u_{ij} | \forall v_i \in A, 1 \leq j \leq \rho_i\}$ such that each $S_j, \forall 1 \leq j \leq t$, has at least $\mathcal{K}$ elements from $\mathcal{U}$. This can be viewed as if there are $\mathcal{K}$ copies

of set S_j given as S_{jk}, $1 \leq k \leq \mathcal{K}$ and set $\mathcal{U}$ hits these sets at least once. So altogether there are $\lambda = \mathcal{K}|\mathcal{S}| = \mathcal{K}t$ sets hit by $\mathcal{U}$. Since the algorithm follows the greedy approach of general hitting set problem, the final approximation ratio of DDA is $O(\ln(\mathcal{K}t))$, *i.e.*, $O(\ln(K) + 2\ln(n+m))$ [68]. □

5.4.3 Experimental Results for Accuracy of Data Distribution Algorithm

Extensive simulations are performed to demonstrate the efficacy of the proposed approximation algorithm (DDA Algorithm 7 for Data Distribution problem, by comparing its solution to the optimal number of storage required in the network. All the experiments are performed on two real fiber backbone networks of a major network provider [60]: (i) USA network (147 nodes) and (ii) Europe network (46 nodes). The (x, y) coordinate of a node on the network layout is taken to be the latitude and longitude of the corresponding city in the map. A fiber link between two cities in the network map is taken as a straight line between the corresponding nodes in the network layout. All distance units in this section are in latitude and longitude coordinates (*i.e.*, one unit is approximately 60 miles).

Two parameters are considered that impact the comparison results - (i) radius of the circular fault region r, and (ii) the number of file segments required to reconstruct the file ($\mathcal{K}$). In all the experiments the storage capacity of each node (measured in terms of number of segments it can store) is taken to be a random value between 1 and $\mathcal{K}/2$.

First, the impact of change in the radius r of circular fault region on the total storage requirement is studied, keeping $\mathcal{K}$, $\mathcal{N}$ and storage capacity of each node constant. In these experiments, $\mathcal{K} = 10$ and $\mathcal{N} = 20$ for both the networks. The radius r is varied from 0.5 unit to 2.5 unit in steps of 0.5 (*i.e.*, from 30 miles

to 150 miles). Figure 5.4 shows the result. In most of the cases DDA solution is equal to the optimal solution and in other cases the difference between the two is less than 2. While the ILP takes several hours to find the optimal solution when r is large DDA takes only a few seconds to find a solution.

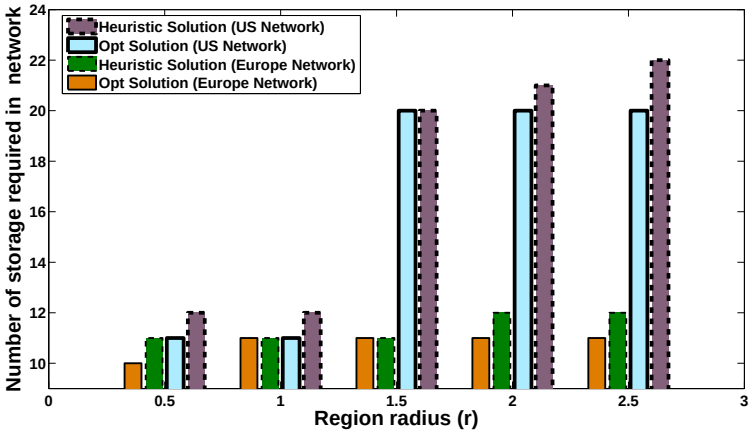


Figure 5.4: Network storage vs Region Radius in a US and Europe fiber network when the value of $\mathcal{K} = 10$

Secondly, the impact of change in the value of $\mathcal{K}$ on the total storage requirement is studied, keeping r and $\mathcal{N}$ constant. The radius r is set to 1.5 unit (90 miles) and $\mathcal{N}$ is set to 20 for both the networks. The parameter $\mathcal{K}$ is varied from 2 to 10 in steps of 2. Figure 5.5 shows the result. In all cases the DDA solution matches exactly with the optimal solution. The increase in the storage requirement in the network with increase in $\mathcal{K}$ is quite intuitive. Higher values of $\mathcal{K}$ implies that a node requires a larger number of distinct file segments to reconstruct the file. This requirement results in higher storage need for the file segments.

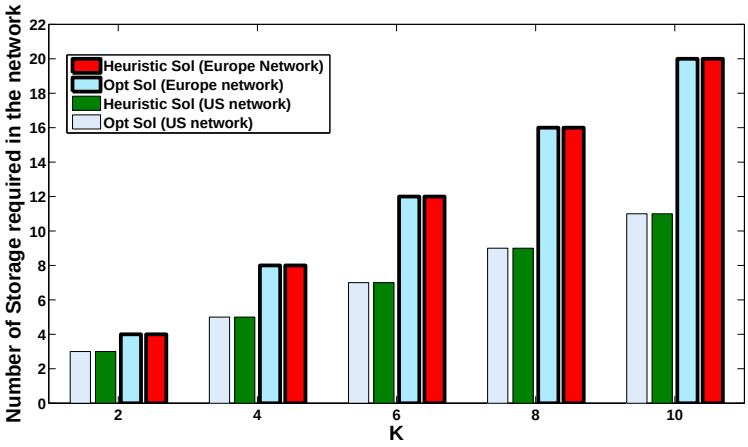


Figure 5.5: Network storage vs $\mathcal{K}$ in a US and Europe fiber network when the value of $r = 1.5$ units

Algorithm 7: Data Distribution Algorithm (DDA)

Input :

1. The layout of graph $G = (V, E)$ on a two dimensional plane $LG = (P, L)$,
2. Maximum α_i colors can be assigned to v_i, $\forall v_i \in V$,
3. Color set $\mathcal{C} = \{c_i, \ldots, c_{\mathcal{N}}\}$,
4. Region radius r,
5. Parameter $\mathcal{K}$ and $\mathcal{N} = \max_j \sum_{v_i \in S_j} \alpha_i$.

Output: Assignment of colors $C_i = \{c_1, \ldots, c_{\rho_i}\}$ to each node v_i, $\forall 1 \leq i \leq n$ such that $C_i \subseteq \mathcal{C}$, $|C_i| = \rho_i \leq \alpha_i$, for any region fault of radius r the largest component of the residual graph G' has at least $\mathcal{K}$ distinct colors and $\sigma = \sum_i^n \rho_i$ is minimum.

Compute all the regions $\mathcal{R} = \{R_1, \ldots, R_t\}$ of radius r in LG using the method described in Chapter 2;
Find the largest component S_i of the residual graph for each region fault $R_i \in \mathcal{R}$. Let $\mathcal{S} = \{S_1, \ldots, S_t\}$;
$\mathcal{Q} = \{\alpha_i, \ldots, \alpha_n\}$;
$A \leftarrow \emptyset$;
$C_i \leftarrow \emptyset, \ \forall i = 1, \ldots, n$;
$k_j = \mathcal{K}, \ \forall j = 1, \ldots, t$;
while $\mathcal{S} \neq \emptyset$ **do**
    Compute $\mathcal{T} = \{H_1, \ldots, H_n\}$ for all $1 \leq i \leq n$, where H_i is the total number of times node v_i appears in the remaining sets of $\mathcal{S}$;
    if $\alpha_p = 0$ *for all* $v_p \in V$ **then**
        No feasible solution exist; **return**;
    Pick $v_p \in V$ with $\alpha_p \neq 0$ such that corresponding H_p is maximum in $\mathcal{T}$;
    $A \leftarrow A \cup v_p$;
    forall the S_j *such that* $v_p \in S_j$ **do**
        Let $D_j \subseteq \mathcal{C}$ be the set of colors available in set S_j such that $D_j \cap C_i = \emptyset$ for all $v_i \in S_j$;
        $k_j \leftarrow k_j - 1$;
        if $k_j = 0$ **then** $\mathcal{S} \leftarrow \mathcal{S} \setminus S_j$;
    Let c be the minimum color in $\bigcap_{S_j : v_p \in S_j} D_j$, then $C_p \leftarrow C_p \cup c$;
    $\alpha_p \leftarrow \alpha_p - 1$;
Let nodes in A are added in the order $\{v^1, v^2, \ldots, v^{|A|}\}$ and ρ_i colors have been assigned to node v^i in the order $C_i = \{c_1, c_2, \ldots, c_{\rho_i}\}$
for $i = |A|$ *to* 1 **do**
    for $j = \rho_i$ *to* 1 **do**
        if $C_i \setminus c_j$ *gives a feasible solution* **then**
            $C_i \leftarrow C_i \setminus c_j$;

Chapter 6

FAULT TOLERANT RESOURCE ALLOCATIONS IN DATA CENTERS

Cloud computing has emerged as an important technique in the way that organizations consume computing and storage resources. Cloud computing provides organizations with the unique ability to scale their computing requirements rapidly to meet the business growth and the occasional spikes in demand, in a cost-effective way. The rapid scaling and the cost effectiveness of the computing resources are made possible through the fundamental technology of *Hardware Virtualization* or *Virtual Machines*. Virtualization allows running multiple Virtual Machines (VMs) on a single physical server, with each VM sharing the resources of that server across multiple environments. Different VMs can run different operating systems and applications, all on the same server. Increasingly, VMs are being hosted in large-scale public data centers, such as Amazon EC2, Microsoft Azure, Google App Engine, that typically house hundreds of thousands of servers. Figure 6.1 shows the high-level design of such a data center.

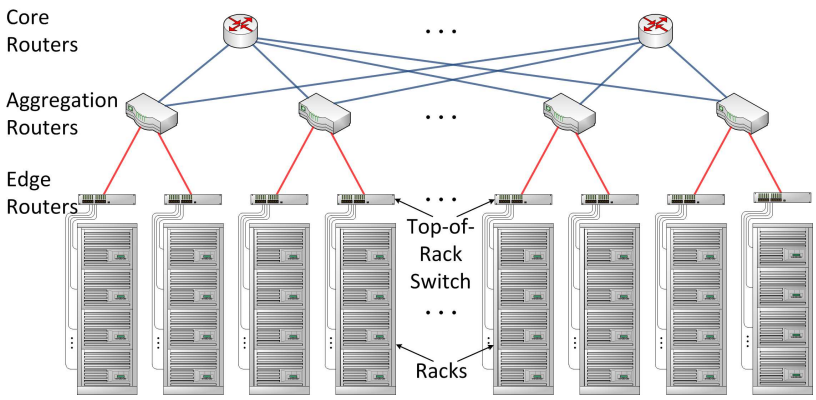


Figure 6.1: A hierarchical 3-tier Data Center Architecture

As shown in the figure, multiple physical servers (henceforth referred as *servers*) are located on a *rack*. In practice, the racks are interconnected through a simple 3-tier hierarchical topology (recently, interconnection topologies with more attractive features have been suggested in [69–71]). Each server is capable of hosting a certain number of VMs. For the purpose of service differentiation, cloud operators typically offer virtual machines in different capacities. For instance, Amazon EC2 [72] offers VMs having different compute (CPU) and memory capacities.These are referred in this thesis as different *types* of VMs. End users request for a certain number of VMs of a particular type from the cloud infrastructure to run their applications. A centralized cloud resource manager software services the user requests and allocates the requested VMs on servers. The resource manager keeps tracks of the resource usage and the health of the VMs in the data center.

A large-scale data center experiences random failures of several hardware components every day. The result of these failures is that one or more VMs allocated to an end user may become unavailable, until the time the VMs are reprovisioned on other servers in the data center. Thus, it becomes extremely critical to allocate VMs for an end user in a way that, even in the presence of failures in the data center, one or more of the user's VMs remain operational. This notion of *Fault Domain*, originally suggested by Microsoft in Azure design [73], helps capture fault-tolerance aspect of VM allocation. A *fault domain* is defined as a set of servers (correspondingly, the set of VMs hosted on them), all of which become unavailable when a single fault occurs in the data center. For instance, if a switch or a router of the data center network fails, then all the servers connected through this switch/router may become inaccessible. The net effect of failure of such a switch/router is equivalent to simultaneous failure of all the

servers connected through this switch/router. Accordingly, this set of servers form a single fault domain. For example, in Figure 6.1, all the servers belonging to a single rack form a fault domain.

From the reliability perspective, an end user requesting a certain number of VMs may want to specify the number of fault domains in which her VMs be distributed. Although in today's public cloud infrastructures, end users cannot make explicit request for fault domains, it is foreseeable that in the near-future, cloud providers will offer this as a way to provide *differentiated quality of service*. If two users U_1 and U_2 request the same number of VMs, but distributed over two different fault domain sizes D_1 and D_2 where $D_1 > D_2$, then the data center operator can charge more to U_1, because in some sense this user is receiving more "fault-tolerant" service than user U_2. Based on the resource availability at the time a user request arrives in the data center, the request may or may not be satisfied. If the request is satisfied, then the user is charged for each VM that is allocated for the duration of time that the user holds the resources and a fixed charge for each fault domain requested by the user.

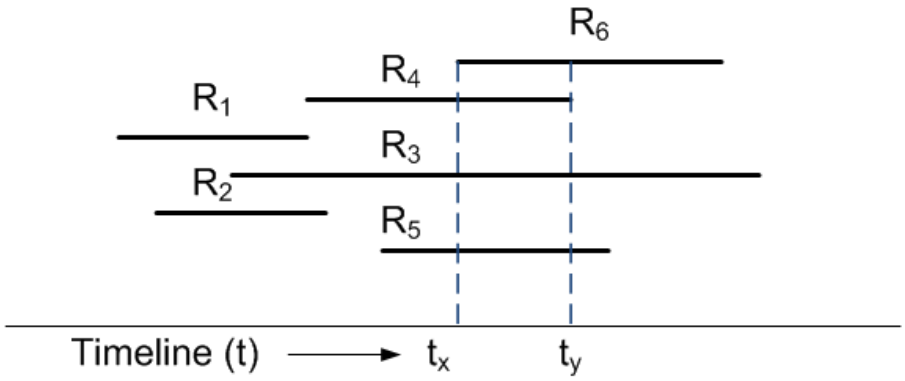


Figure 6.2: Timeline of the resource requests to the data center

In practice, it is difficult to predict the resource requirements for cloud computing applications [74]. The dynamic user requests for resources can ar-

rive at any time and stay for any amount of time. A request arrival pattern in a data center is shown in Figure 6.2. The requests R_1 through R_6 arrive at different points of time with different VM and fault domain requirements. It may be noted from the figure that during the time interval t_x to t_y, four requests R_3, R_4, R_5, R_6 are active - *i.e.*, they are requesting resources during that time period. In this thesis, a similar time interval, where a number of requests are active, is considered. In this research an algorithm is presented to help the data center operator to decide which requests have to be satisfied, with the available set of resources (VMs and fault domains) in that time period, in order to maximize its revenue. It may be noted that if the arrival/departure time of all the requests were identical, it would be equivalent to focusing on the interval $t_x - t_y$. This simplification serves two main purposes. Firstly, the solution computed for this simplified problem serves as a benchmark for online allocation techniques that solve the problem for the dynamic requests case. Secondly, it facilitates the design of efficient solution techniques, which can be adapted to solve the dynamic request problem. Further, this is a reasonable assumption, since there are many works in literature that consider historical workload request traces to predict future workload demands [75, 76] and the predicted information is taken as apriori request information to make the resource allocation decisions.

Another assumption that is made is that each server can host VMs of a single type. Although each server is capable of running VMs of different types, in practice, the data center operators partition the servers among virtual machine types for the ease of maintenance. As such, the solution techniques proposed in this thesis can be extended to accommodate the case, in which a server hosts VMs of different types. The contributions of this research are as follows $-$

- Provide novel formulation of the fault-tolerant data center resource alloca-

tion problem, taking into account the fault domains in the data center.

- Prove that this problem is NP-complete and provide an Integer Linear Program formulation to compute the optimal solution.
- Provide an efficient heuristic that produces near-optimal solution in a fraction of time required to compute the optimal solution.
- Through extensive experimentation, prove the efficacy of this heuristic.

6.1 Related Work

The vast collection of research literature that solves the problem of resource allocation in data centers can be categorized into two groups. The first group of research papers address the problem of optimal resource usage while maintaining application performance guarantees [77–79]. In [77], the authors define the problem of adaptive resource allocation in data center as a non-linear continuous optimization problem and provide a framework and efficient techniques to solve the problem. In [78], the authors have proposed a global resource scheduling algorithm that provides on-demand capacities to the hosted services using the notion of resources flowing among VMs. The second group of research papers, in this domain, addresses the problem of temperature-aware resource allocation to optimize the power consumption in the data center [80–82]. In [80], the authors have proposed a control-theoretic approach to solve the optimal resource allocation and power management problem in data centers with dynamic workloads. In [81, 82], the authors have proposed temperature-aware resource allocation schemes to minimize peak temperature in a data center. Unlike the existing research literature on data center resource allocation techniques, to the best of our knowledge, this is the *first work on solving the problem*

of fault-tolerant data center resource allocation taking into account the fault domains.

6.2 System Model and Problem Formulation

System model comprises of a set of *Physical Servers*, $P = \{P_1, P_2, \ldots, P_n\}$ capable of hosting *Virtual Machines (VM)*. A VM can be of m different types, $T = \{T_1, T_2, \ldots, T_m\}$ and each server can host multiple VMs of a single type. Each element $n_j, 1 \leq j \leq m$, of the vector $N = \{n_1, n_2, \ldots, n_m\}$ indicates the *number* of virtual machines of type T_j present in the system. The set of virtual machines present in the system is represented as $V = \{v_{11}, \ldots, v_{1n_1}, v_{21}, \ldots, v_{2n_2}, \ldots, v_{m1}, \ldots, v_{mn_m}\}$, where each VM $v_{ij} \in V$ represents jth virtual machine of type T_i. A server P_j hosting VMs of type T_i will have virtual machines $V'_j \subseteq \{v_{i1}, \ldots, v_{in_i}\}$, $\forall 1 \leq j \leq n$. Total number of VMs present in the system is given by $\rho = \sum_{i=1}^{m} n_i$.

In the revenue maximization problem considered in this research, it is assumed that the set of all fault domains $F = \{F_1, F_2, \ldots, F_r\}$ is provided as an input of the problem. It may be noted that $F_k \subseteq P$, $\forall 1 \leq k \leq r$, and since each server P_j hosts of a set of virtual machine $V_j \subseteq V$, F_k can be presented as a subset of V. The set of user requests for the resources (virtual machines) of the data center is denoted by the vector $R = \{R_1, R_2, \ldots, R_t\}$. Each request $R_l, 1 \leq l \leq t$, is a *triplet* $< \alpha_l, \beta_l, \gamma_l >$, where α_l, β_l represents the *type* and the *number* of virtual machines and γ_l represents *number of fault domains* requested. If the data center operator is able to allocate resources to satisfy a user request, the operator generates a revenue for the data center. The revenue depends on (i) the number and type of VMs allocated and (ii) the number of fault domains requested. Assuming that $C(\alpha_l)$ is the unit cost of a VM

of type α_l and D is a fixed cost associated with each fault domain, the revenue generated by successful allocation of request R_l is $P(R_l) = \beta_l \times C(\alpha_l) + \gamma_l \times D$. The goal of the revenue maximization problem is to satisfy the resource and domain requirements of a subset $R' \subseteq R$ of requests that maximizes the data center revenue. The revenue maximization problem is formally stated as follows.

Problem 8 (Data Center Resource Allocation (*DCRA*))**.** INSTANCE: *Given*
(i) A set of Physical Servers $P = \{P_1, P_2, \ldots, P_n\}$*,*
(ii) A set of virtual machines
$V = \{v_{11}, \ldots, v_{1n_1}, v_{21}, \ldots, v_{2n_2}, \ldots, v_{m1}, \ldots, v_{mn_m}\}$*,*
(iii) A set of fault domains $F = \{F_1, F_2, \ldots, F_r\}$*, where* $F_k \subseteq P$*,* $\forall 1 \leq k \leq r$*. (Since each element of* P *represents a subset of* V*,* $F_k \subseteq V$*,* $\forall 1 \leq k \leq r$*),*
(iv) A set of user requests $R = \{R_1, R_2, \ldots, R_t\}$*, where each request* $R_l, 1 \leq l \leq n$*, is a* triplet $< \alpha_l, \beta_l, \gamma_l >$*, where* $< \alpha_l, \beta_l, \gamma_l >$ *are as defined earlier,*
(v) $C(\alpha)$*: the unit cost of a VM of type* α*,* $\forall \alpha \in T$*,*
(vi) D*: Fixed cost associated with each fault domain,*

Objective*: To find a subset* $R' \subseteq R$*, such that the resource requirement of all requests in* R' *is satisfied and the revenue* $P(R')$ *generated by* R' *is maximum.*

6.3 Optimal Solution with Integer Linear Program

It can be proved that *DCRA* problem is NP-Complete by a transformation from the *Subset Sum problem* which is known to be NP-Complete [42]. The proof can be found in Appendix B. In this section, methods are proposed to solve the problem optimally.

An Integer Linear Program (ILP) is formulated to solve the DCRA problem optimally. For each VM, $v_{ij} \in V$ and each fault domain F_k, $\forall\{F_k|v_{ij} \in F_k\}$, and for each request R_l, $\forall\{R_l|\alpha_l = T_i\}$, a binary variable v_{ij}^{kl} is introduced. v_{ij}^{kl} is equal to 1 if and only if the VM v_{ij} is used for the lth user request in the kth fault domain. Also for each fault domain F_k, and for each request R_l, a variable f_k^l is used. f_k^l is equal to 1 if and only if at least one VM is allocated from fault domain F_k for the lth request. For each user request R_l, $\forall 1 \leq l \leq t$, a variable η_l is used, which is equal to 1 if and only if user request R_l is satisfied. The objective is to allocate virtual machines for the requests in such a way that the subset of requests with maximum revenue is satisfied.

Constraint 1 is the *resource constraint* of a request and ensures that if a request R_l is satisfied then β_l number of VMs of type α_l are allocated for that request. On the other hand, constraint 2 is the *fault-domain constraint*, which ensures that if a request R_l is satisfied, then the VMs allocated for the request span γ_l fault domains. Constraint 3 ensures that a virtual machine v_{ij} can only be allocated to a single request and in a single fault domain. Constraint 4 implies that if at least one virtual machine v_{ij} is allocated for the request R_l in fault domain F_k, then f_k^l will be 1. Note that since f_k^l is binary variable, it can be at most 1.

Variables: For each node v_{ij} and fault domain F_k, $\forall\{F_k|v_{ij} \in F_k\}$, and for each request R_l, $\forall\{R_l|\alpha_l = T_i\}$

$$v_{ij}^{kl} = \begin{cases} 1, & \text{if VM } v_{ij} \text{ is used for the } R_l \text{ in the } F_k \\ 0, & \text{otherwise.} \end{cases}$$

For each fault domain F_k and request R_l,

$$f_k^l = \begin{cases} 1, & \text{if at least one VM is allocated from } F_k \\ & \text{for the request } R_l \\ 0, & \text{otherwise.} \end{cases}$$

For each user request R_l,

$$\eta_l = \begin{cases} 1, & \text{if user request } R_l \text{ is satisfied} \\ 0, & \text{otherwise.} \end{cases}$$

ILP:

$$\max \quad \sum_{l=1}^{t} P(R_l)\eta_l$$

$$s.t \quad \sum_{k:v_{ij}\in F_k} \sum_{j=1}^{n_i} v_{ij}^{kl} \geq \eta_l \beta_l, \quad \begin{cases} \forall l = 1, \ldots, t \\ \text{and } \{i | \alpha_l = T_i\} \end{cases} \tag{6.1}$$

$$\sum_{k}^{r} f_k^l \geq \eta_l \gamma_l, \quad \forall l = 1, \ldots, t \tag{6.2}$$

$$\sum_{k:v_{ij}\in F_k}^{r} \sum_{l:\alpha_l = T_i} v_{ij}^{kl} \leq 1, \quad \begin{cases} \forall i = 1, \ldots, m, \\ \forall j = 1, \ldots, n_i \end{cases} \tag{6.3}$$

$$f_k^l \leq \sum_{\substack{v_{ij}\in F_k \\ i:\alpha_l = T_i}} v_{ij}^{kl}, \quad \forall l = 1, \ldots, t, \ k = 1, \ldots, r \tag{6.4}$$

6.4 Heuristic Solution Using a Greedy Approach

In this section, a heuristic (RAA) (Algorithm 8) is presented for the DCRA problem. The inputs for the algorithm can be partitioned for different types of VMs present in the system. The set $\mathcal{V} = \{V_{T_1}, V_{T_2}, \ldots, V_{T_m}\}$ represents the set of m subsets of VMs present in the system, where $V_{T_i} \subseteq V$ is the set of VMs of type T_i, $\forall T_i \in T$. If VM of type T_i is present in r_i fault domains, then

$F_{T_i} = \{F_{T_i,1}, F_{T_i,2}, \ldots, F_{T_i,r_i}\}$ represents set of fault domains containing VMs of type T_i, $\forall T_i \in T$. Each fault domain $F_{T_i,k_i} \subseteq V_{T_i}$, $\forall 1 \leq k_i \leq r_i$, contains VMs only of type T_i. Similarly the request set R is divided into subsets of requests R_{T_i}, $\forall T_i \in T$, where $R_{T_i} = \{R_{T_i,1}, R_{T_i,2}, \ldots, R_{T_i,t_i}\}$ consists of all the requests of VMs of type T_i.

It should be noted that in order to satisfy the requests in R_{T_i} one only needs to consider VMs of set V_{T_i} and fault domains of set F_{T_i} present in the system. The resource allocation problem can be solved for each of these partitioned request sets R_{T_i} independently. The i-th iteration of the *for-loop* in step 2 considers resource allocation of all requests for VMs of type T_i. The *while-loop* at step 3 terminates only when all the requests for VMs of type T_i have been considered or when the resource constraint and fault domain constraint for all the remaining requests can no longer be satisfied. In each iteration within the *while-loop*, one request is selected, among all the remaining requests R_{T_i} of type T_i for resource allocation. For each request $R_{T_i,l} \in R_{T_i}$, *revenue per unit resource* of the request $\varphi(R_{T_i,l})$ is computed (step 4) in the following way,

$$\varphi(R_{T_i,l}) = \frac{C(T_i)\beta_l + D\gamma_l}{\frac{\beta_l}{n_{T_i}} + \frac{\gamma_l}{\Gamma}}$$

where n_{T_i} and Γ respectively denote the number of VMs and the number of fault domains containing VMs of type T_i available in the system. Algorithm 9 describes how to compute φ for a request. In step 4, a request $R_{T_i,l}$ with maximum φ is selected. In step 5, all the fault domains $F_{T_i,k_i} \in F_{T_i}$ are sorted in descending order of their size so that the fault domain with maximum available VM is at the top of the set F_{T_i}. Step 6 checks if request $R_{T_i,l}$ is satisfiable. If the total number of free VMs of type T_i or the total number of fault domains available in the system are less than the demand, then the request is considered unsatisfiable and is removed from the request list.

If the request is satisfiable then the request $R_{T_i,l} =< T_i, \beta_l, \gamma_l >$ is added in the solution set R' (step 9). Also for the request $R_{T_i,l}$, one VM is allocated from each of the first γ_l fault domains in set F_{T_i} (step 11- step 14). This ensures that the fault domain constraint of the request is satisfied. Let $\hat{r}$ be the minimum number of VM present in any fault domain $F_{T_i,k_i} \in F_{T_i}$ after that. The rest of the $\beta_l - \gamma_l$ VMs requested in $R_{T_i,l}$ are allocated in such a way that after the allocation the value of $\hat{r}$ decreases by minimum amount. *While-loops* at step 16 and 20 ensures the same. Variable p counts the number of VM allocated for request $R_{T_i,l}$. When $p \geq \beta_l$, the request $R_{T_i,l}$ is considered to be satisfied and removed from the request set (step 24). Whenever a VM is allocated from a fault domain F_{T_i,k_i}, the VM is removed from the set F_{T_i,k_i}. If in the process of allocation all the VMs in a fault domain F_{T_i,k_i} are used up, *i.e.*, $|F_{T_i,k_i}| = 0$, then F_{T_i,k_i} is removed from the set F_{T_i}.

Theorem 9. *The time complexity of RAA (Algorithm 8) is $O(\rho t)$, where ρ is the total number of VMs in the system and t is the total number of requests.*

Proof. The outer *for-loop* at step 2 runs m times for each type of VM present in the system. The *while-loop* at step 3 runs $\min(t_i, n_i)$ times, where t_i is the number of requests of type T_i and n_i is the number of VMs in system of type T_i. Finding the request of maximum φ at step 4 will take $O(t_i)$ time. Sorting the fault domains in step 5 will take $O(r_i \log r_i)$ time, where r_i is the number of fault domains containing VMs of type T_i. If the request is satisfiable then all the virtual machines will be allocated to the request in $O(\beta_l)$ time (step 11 to 20). Altogether the complexity of the algorithm is $O(\sum_{i=1}^{m} \min(t_i, n_i)(t_i + r_i \log r_i + \beta_l))$. In general, the number of requests t_i will be greater than r_i, β_l

and n_i. So the time complexity of the Algorithm 8 is $O(\sum_{i=1}^{m} n_i t_i)$ or $O(\rho t)$, where $\rho = \sum_{i=1}^{m} n_i$ and $t = \sum_{i=1}^{m} t_i$. □

6.5 Experimental Results

In this section, extensive simulation results are presented to demonstrate the efficacy of the resource allocation heuristic (Algorithm 8) by comparing heuristic solution to the optimal solution. The optimal solutions were obtained by solving ILP presented in Section 6.3 using CPLEX Optimizer 10.0.

Three parameters that impact the comparison results are considered – (i) the number of servers n, (ii) the number of fault domains r available in the system and (ii) the number of user requests t. In all simulations, the hierarchical interconnection pattern of the servers and routers is used as the topology of the data center (Figure 6.1). Fault domains are computed based on this topology. Note that in this topology, fault domains consist of racks of servers that becomes isolated due to the failure of any aggregation router. All other fault domains (formed due to failure of a single server or a single edge router) will be a subset of the fault domain formed due to the failure of any aggregation router and so it is not required to consider these fault domains separately. Details of the experimental setup are presented in Table 6.1. The pricing for VM usage is obtained from the pricing charges of Amazon EC2 virtual machines with windows usage in US East (Virginia) region [72].

In the first set of experiments, the impact of change in the number of servers n in the data center is studied on the performance of the heuristic, while keeping other parameters constant. In these experiments, the number of requests and available fault domains are fixed at 1000 and 8 respectively. The number of servers is taken to be 50, 100, 200, 300 and 400. For each value of n,

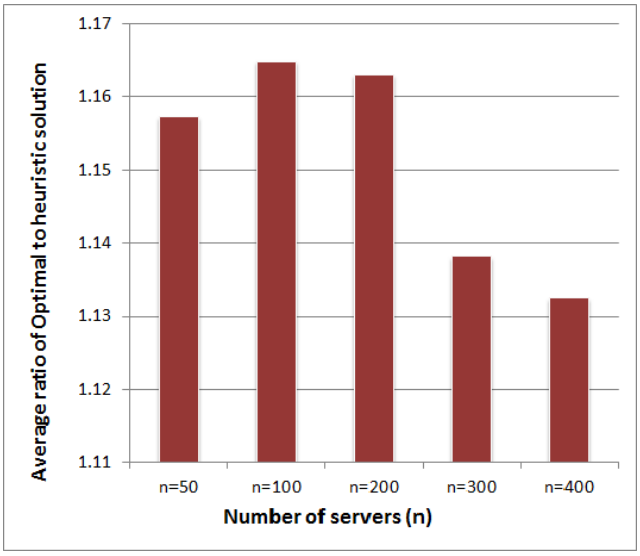


Figure 6.3: Experimental results: Average ratio of the Optimal to the Heuristic solution for different number of servers.

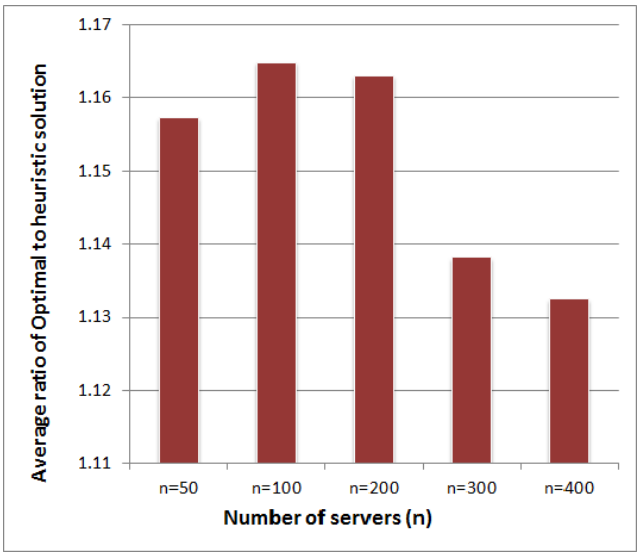


Figure 6.4: Experimental results: Average ratio of the Optimal to the Heuristic solution for different number of fault domains.

experiments are ran for 5 different instances. The average ratio of the optimal to heuristic solution, for all the runs, is presented in Figure 6.3.

In the second set of experiments, keeping other parameters constant, the impact of change in number of fault domains r available in system is studied on

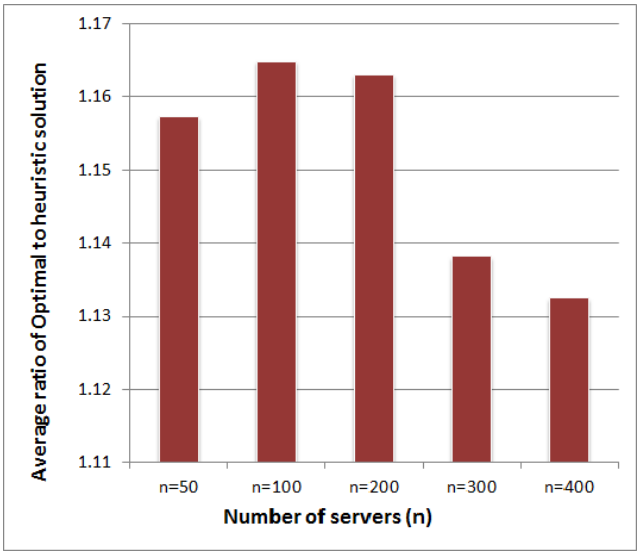


Figure 6.5: Experimental results: Average ratio of the Optimal to the Heuristic solution for different number of requests.

the performance of the heuristic. In these experiments, the number of requests and available servers are fixed at 1000 and 100 respectively. The number of available fault domains in the system is varied from 4 to 12 in steps of 2. For each value of r, experiments are ran for 5 different instances. The average ratio of the optimal to heuristic solution for all the runs is presented in Figure 6.4. The ratio of optimal to heuristic solution increases with increase in r but is at most 1.18.

In the third set of experiments, experiments are repeated for different number of requests t, keeping all other parameters unchanged. In these experiments, the number of available servers and fault domains are fixed to 100 and 8 respectively. The number of requests is varied from 500 to 2500 in steps of 500. For each value of t, experiments are ran for 5 different instances. The average ratio of the optimal to heuristic solution is presented in Figure 6.5.

In all these experiments, the ILP typically took several hours (3-4 hours) to compute the optimal solution for each instance. However, the heuristic com-

puted the near-optimal solution within a fraction of a second in all cases. Also in all the cases the heuristic solution is always within a factor of 1.2 of the optimal solution.

Algorithm 8: Resource Allocation Algorithm (RAA)

Input :

1. A set of subsets of virtual machines $\mathcal{V} = \{V_{T_1}, V_{T_2}, \ldots, V_{T_m}\}$, where $V_{T_i} \subseteq V$ is the set of VMs of type T_i, $\forall 1 \leq i \leq m$, available in the system,
2. A set of subsets of fault domains $\mathcal{F} = \{F_{T_1}, F_{T_2}, \ldots, F_{T_m}\}$, where $F_{T_i} = \{F_{T_i,1}, F_{T_i,2}, \ldots, F_{T_i,r_i}\} \subseteq F$ and each $F_{T_i,k_i} \subseteq V_{T_i}$, $\forall 1 \leq k_i \leq r_i$ and $1 \leq i \leq m$,
3. A set of subsets of requests $\mathcal{R} = \{R_{T_1}, R_{T_2}, \ldots, R_{T_m}\}$, where $R_{T_i} = \{R_{T_i,1}, R_{T_i,2}, \ldots, R_{T_i,t_i}\} \subseteq R$ is the set of requests for VMs of type T_i, $\forall 1 \leq i \leq m$,
4. $C(\alpha)$ is the cost of using one VM of type $\alpha \in T$ and D is the fixed cost associated with each fault domain.

Output : A subset of requests $R' \subseteq R$ whose VMs and fault domain requirements can be satisfied and that maximizes the total revenue.

$R' \longleftarrow \emptyset$
for $i \longleftarrow 0$ **to** m **do**
  while $R_{T_i} \neq \emptyset$ *and* $F_{T_i} \neq \emptyset$ **do**
    Find the request $R_{T_i,l} \in R_{T_i}$ with maximum efficiency φ using the Algorithm 9. Let $R_{T_i,l}$ is given as $< T_i, \beta_l, \gamma_l >$
    Sort the sets $F_{T_i,k_i} \in F_{T_i}$ in descending order of their size
    if $|F_{T_i}| < \gamma_l$ *or* $\sum_{F_{T_i,l} \in F_{T_i}} |F_{T_i,l}| < \beta_l$ **then**
      $R_{T_i,l}$ can not be satisfied
      $R_{T_i} \longleftarrow R_{T_i} \setminus R_{T_i,l}$; Go To step 3
    $R' \longleftarrow R' \cup R_{T_i,l}$
    $p \longleftarrow 0$
    for $k \longleftarrow 1$ **to** γ_l **do**
      Allocate one VM from $F_{T_i,k}$ for request $R_{T_i,l}$; Set $p \leftarrow p + 1$
      Remove the VM from set $F_{\alpha_i,k}$
      if $|F_{T_i,k_i}| = 0$ **then** $F_{T_i} \longleftarrow F_{T_i} \setminus F_{T_i,k_i}$
    Let $\hat{r} = \min_{F_{T_i,k_i} \in F_{T_i}} |F_{T_i,k_i}|$
    while $p < \beta_l$ *and* $\exists F_{T_i,k_i} \in F_{T_i}$ *such that* $|F_{T_i,k_i}| > \hat{r}$ **do**
      Allocate one VM from next $\{F_{T_i,k_i} \in F_{T_i} \mid |F_{T_i,k_i}| > \hat{r}\}$ for request $R_{T_i,l}$ in a round-robin fashion
      Remove that VM from set F_{T_i,k_i}; $p \leftarrow p + 1$
      if $|F_{T_i,k_i}| = 0$ **then** $F_{T_i} \longleftarrow F_{T_i} \setminus F_{T_i,k_i}$
    while $p < \beta_l$ **do**
      Allocate one VM from next $F_{T_i,k_i} \in F_{T_i}$ for request $R_{T_i,l}$ in a round-robin fashion;
      Remove that VM from set F_{T_i,k_i}; $p \leftarrow p + 1$
      if $|F_{T_i,k_i}| = 0$ **then** $F_{T_i} \longleftarrow F_{T_i} \setminus F_{T_i,k_i}$
    $R_{T_i} \longleftarrow R_{T_i} \setminus R_{T_i,l}$
return R'

Algorithm 9: Finding $\varphi(R)$ for the request R

Let $R =< \alpha, \beta, \gamma >$
$n_\alpha \longleftarrow$ number of available VMs of type α
$\Gamma \longleftarrow$ number of available fault domains containing VMs of type α
Revenue rev $\longleftarrow C(\alpha)\beta + D\gamma$
Required Resources res $\longleftarrow \frac{\beta}{n_\alpha} + \frac{\gamma}{\Gamma}$
return $\frac{rev}{res}$

Table 6.1: Parameter values for Simulation

Parameter	**Values**
Topology	Hierarchical
Number of nodes	50 to 400
Number of VM per node	2 to 8
Types of VM	3
Price of each type of VM	$\$0.12$, $\$0.48$, $\$0.96$
Price associated with one fault domain	$\$0.12$
Number of fault domains	4, 6, 8, 10 and 12
Number of requests	500 to 2500
Number of VM requested in each request	1 to 8
Number of fault domain requested in each request	1 to 8

Chapter 7

CONCLUSIONS AND FUTURE RESEARCH DIRECTIONS

7.1 Conclusions

Spatially correlated failures or *region-based* failures, induced by natural calamities or human errors, are very common form of network failures in present world. With the huge progress in computer networks, it has also become extremely critical to ensure a high level of fault tolerance capability in the networks against such region-based node and link failures. In presence of such region-based faults, many of traditional network analysis and fault-tolerant metrics, that are valid under non-spatially correlated faults, are no longer applicable. As for example, maximum number of region-disjoint paths is no longer equal to minimum region-cut in complete region fault model. Also, connectivity is not the most appropriate fault-tolerant metric for resource efficient design of networks in presence of region-based faults. To this effect, the the principal focus of this thesis is the study of design and analysis of robust networks in presence of such region-based faults. Insight of this research is that if some prior knowledge is available on the maximum size of the region that might be affected due to a region-based fault, this piece of knowledge can be effectively utilized for resource efficient design of networks. This thesis introduces several new metrics - region-based connectivity, region-based component decomposition number (RBCDN), region-based largest component size (RBLCS) and region-based smallest component size (RBSCS), appropriate for measuring the robustness of network in presence of region-based faults. This thesis also proposes efficient design techniques with these new metrics, suitable for designing communication networks in presence of region-based faults. Moreover, this thesis presents

efficient resource allocation techniques in two specific networks, data storage network and data center network, in presence of region-based faults.

7.2 Future research directions

Research in designing and analyzing networks, in presence of spatially correlated faults, is a relatively nascent area of study and offers several unsolved problems. This area of study has received considerable attention in present network research community. This thesis attempts to address and solve some of the important problems and in process opens up new venues of research. This section discusses some of the open problems that need to be addressed for future research.

- This thesis concentrates mainly on region-based node and link failures that are deterministic in nature, *i.e.*, failure probability of nodes and links of a network outside the fault area is zero. In a more realistic setting the failure of nodes and links are probabilistic in nature, *i.e.*, probability of a node and link failure decreases with the increase in distance from the epicenter of region-fault. Although some works [19,25,27] have been done in accessing reliability and capacity degradation of a network in presence of probabilistic region-based faults, few important questions remained unanswered, *e.g.*,

    1. Is connectivity or region-based connectivity still a good metric of measuring fault tolerance in presence of probabilistic region-based failures?
    2. If not, what will be appropriate metrics to capture network robustness under such model?

3. How to design and analyze networks under probabilistic region-based fault model with these metrics?

- This thesis showed that even though Menger's Theorem is valid in general graphs, it is no longer valid when region-based faults are considered, *i.e.*, max-flow is no longer equal to min-cut under geometric region-based failure model. In a recent research [28], authors conjectured that difference between max-flow and min-cut is at most one, *i.e.*, $max - flow \leq min - cut \leq max - flow + 1$. However, a formal proof of the relation between max-flow and min-cut is still an open research problem. Also, it will be interesting to investigate the behavior of other classical graph results under region-based failure model.

- This thesis proposes novel metrics like region-based component decomposition number (RBCDN), region-based largest component size (RBLCS) and region-based smallest component size (RBSCS), to measure the robustness of a network after it gets disconnected due a region-based fault. This thesis also presented network design techniques with a target value of RBCDN and RBLCS. Immediate extension of this research will be as follows :

  1. How to come up with efficient network design techniques based on RBSCS metric?
  2. Improvement in the proposed design techniques of RBCDN and RBLCS.

- This thesis proposes efficient resource allocation techniques in data storage networks and data center networks, in presence of region-based faults. Several possible future research problems in this setup are as follows:

1. Finding out more robust network topology inside data centers, robust against region-based failures.
2. Improving resource allocation techniques in data center considering the network topology, fault-domains and network bandwidth.
3. Finding out efficient data distribution schemes in data storage network for more specific topology, *e.g.*, grid, tree, etc.

Research in this thesis makes some important contributions in this new field of network robustness in presence of spatially correlated faults or region-based faults. However, this area in computer networks literature has a wide potential that yet need to be investigated and researched.

REFERENCES

[1] D. Masi, E. Smith, and M. Fischer, “Understanding and mitigating catastrophic disruption and attack,” *Sigma Journal*, pp. 16–22, 2010.

[2] Y. Xu and W. Wang, “Characterizing the spread of correlated failures in large wireless networks,” in *INFOCOM, 2010 Proceedings IEEE*. IEEE, 2010, pp. 1–9.

[3] N. Ahmed, S. Kanhere, and S. Jha, “The holes problem in wireless sensor networks: a survey,” *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 9, no. 2, pp. 4–18, 2005.

[4] A. Sen, B. Shen, L. Zhou, and B. Hao, “Fault-tolerance in sensor networks: A new evaluation metric,” in *INFOCOM*, 2006.

[5] M. D. Penrose, “On k-connectivity for a geometric random graph,” *Wiley*, vol. 15, no. 2, pp. 145–164, 1999.

[6] ——, “the longest edge of the random minimal spanning tree,” *The Annals of Applied Probability*, vol. 7, no. 2, pp. 340–361, 1997.

[7] F. Xue and P. R. Kumar, “The number of neighbors for connectivity of wireless networks,” *Wireless Networks*, vol. 10, no. 2, pp. 169–181, March 2004.

[8] M. J. B. Appel and R. P. Russo, “The connectivity of a graph on uniform points on $[0, 1]^d$,” *Statistics & Probability Letters*, vol. 60, pp. 351–357, 2002.

[9] P. Gupta and P. R. Kumar, “Critical power for asymptotic connectivity in wireless networks,” in *Stochastic Analysis, Control, Optimization and Applications*, 1998, pp. 547–566.

[10] C. Bettstetter, “On the connectivity of ad hoc networks,” *The Computer Journal*, vol. 47, no. 4, pp. 432–447, 2004.

[11] X.-Y. Li, P.-J. Wan, Y. Wang, and C.-W. Yi, “Fault tolerant deployment and topology control in wireless networks,” in *MobiHoc ’03: Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing*, 2003, pp. 117–128.

[12] K. Eswaran and R. Tarjan, “Augmentation problems,” *SIAM Journal on Computing*, vol. 5, p. 653, 1976.

[13] G. Frederickson and J. JaŠJa, “Approximation algorithms for several graph augmentation problems,” *SIAM Journal on Computing*, 1981.

[14] T. Dinh, Y. Xuan, M. Thai, P. Pardalos, and T. Znati, “On new approaches of assessing network vulnerability: Hardness and approximation,” *Networking, IEEE/ACM Transactions on*, vol. 20, no. 2, pp. 609 –619, april 2012.

[15] M. Bakkaloglu, J. J. Wylie, C. Wang, and G. R. Ganger, “On correlated failures in survivable storage systems,” Carnegie Mellon University, Tech. Rep. CMU-CS- 02-129, 2002.

[16] S. Banerjee, S. Shirazipourazad, and A. Sen, “On region-based fault tolerant design of distributed file storage in networks,” in *INFOCOM, 2012 Proceedings IEEE*, march 2012, pp. 2806 –2810.

[17] W. Cui, I. Stoica, R. H. Katz, and Y. H. Katz, “Backup path allocation based on a correlated link failure probability model in overlay networks,” in *10th IEEE ICNP*, 2002.

[18] S. Nath, H. Yu, P. B. Gibbons, and S. Seshan, “Tolerating correlated failures in wide-area monitoring services,” Intel Corporation, Tech. Rep. IRP-TR-04-09, 2004.

[19] J. Liu, X. Jiang, H. Nishiyama, and N. Kato, “Reliability assessment for wireless mesh networks under probabilistic region failure model,” *Vehicular Technology, IEEE Transactions on*, vol. 60, no. 5, pp. 2253–2264, 2011.

[20] A. Sen, S. Murthy, and S. Banerjee, “Region-based connectivity-a new paradigm for design of fault-tolerant networks,” in *Proceedings of IEEE HPSR, 2009*, 2009, pp. 1–7.

[21] J. Fan, T. Chang, D. Pendarakis, and Z. Liu, “Cost-effective configuration of content resiliency services under correlated failures,” in *Proceedings of the International Conference on DSN*, 2006.

[22] S. Banerjee, S. Murthy, and A. Sen, “On a fault-tolerant resource allocation scheme for revenue maximization in data centers,” in *Advanced Net-*

works and Telecommunication Systems (ANTS), 2011 IEEE 5th International Conference on, dec. 2011, pp. 1 –6.

[23] S. Neumayer, G. Zussman, R. Cohen, and E. Modiano, "Assessing the vulnerability of the fiber infrastructure to disasters," in *INFOCOM 2009, IEEE*. IEEE, 2009, pp. 1566–1574.

[24] S. Neumayer and E. Modiano, "Network reliability with geographically correlated failures," in *INFOCOM, 2010 Proceedings IEEE*. IEEE, 2010, pp. 1–9.

[25] P. Agarwal, A. Efrat, S. Ganjugunte, D. Hay, S. Sankararaman, and G. Zussman, "The resilience of wdm networks to probabilistic geographical failures," in *INFOCOM, 2011 Proceedings IEEE*. IEEE, 2011, pp. 1521–1529.

[26] H. Yu, C. Qiao, V. Anand, X. Liu, H. Di, and G. Sun, "Survivable virtual infrastructure mapping in a federated computing and networking system under single regional failures," in *Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE*. IEEE, 2010, pp. 1–6.

[27] X. Wang, X. Jiang, and A. Pattavina, "Assessing network vulnerability under probabilistic region failure model," in *High Performance Switching and Routing (HPSR), 2011 IEEE 12th International Conference on*. IEEE, 2011, pp. 164–170.

[28] S. Neumayer, A. Efrat, and E. Modiano, "Geographic max-flow and min-cut under a circular disk failure model," in *INFOCOM, 2012 Proceedings IEEE*. IEEE, 2012, pp. 2736–2740.

[29] S. Neumayer, G. Zussman, R. Cohen, and E. Modiano, "Assessing the vulnerability of the fiber infrastructure to disasters," *Networking, IEEE/ACM Transactions on*, vol. 19, no. 6, pp. 1610–1623, 2011.

[30] T. Feyessa and M. Bikdash, "Group-centrality and impact of large-scale localized faults in spatial networks," in *Southeastcon, 2012 Proceedings of IEEE*, march 2012, pp. 1 –5.

[31] D. Bienstock, "Some generalized max-flow min-cut problems in the plane," *Mathematics of operations research*, pp. 310–333, 1991.

[32] A. Sen, S. Banerjee, P. Ghosh, and S. Shirazipourazad, “Impact of region-based faults on the connectivity of wireless networks,” in *Communication, Control, and Computing, 2009. Allerton 2009. 47th Annual Allerton Conference on*, 30 2009-oct. 2 2009, pp. 1430 –1437.

[33] S. Banerjee and A. Sen, “Impact of region-based faults on the connectivity of wireless networks in log-normal shadow fading model,” in *Communications (ICC), 2011 IEEE International Conference on*, june 2011, pp. 1 –6.

[34] S. Banerjee, S. Shirazipourazad, P. Ghosh, and A. Sen, “Beyond connectivity - new metrics to evaluate robustness of networks,” in *High Performance Switching and Routing (HPSR), 2011 IEEE 12th International Conference on*, 2011, pp. 171 –177.

[35] S. Banerjee, S. Shirazipourazad, and A. Sen, “Design and analysis of networks with large components in presence of region-based faults,” in *Communications (ICC), 2011 IEEE International Conference on*. IEEE, 2011, pp. 1–6.

[36] A. Vigneron, “Geometric optimization and sums of algebraic functions,” 2009.

[37] S. Basu, R. Pollack, and M. Roy, “On Computing a Set of Points Meeting Every Cell Defined by a Family of Polynomials on a Variety* 1,” *Journal of Complexity*, vol. 13, no. 1, pp. 28–37, 1997.

[38] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “Wireless Sensor Networks: a Survey,” *Computer Networks*, vol. 38, pp. 393–422, 2002.

[39] R. Diestel, *Graph Theory*. Springer, 2003.

[40] C. Patvardhan, V. Prasad, and V. P. Pyara, “Vertex cutsets of undirected graphs,” *IEEE Transactions on Reliability*, 1995.

[41] T. H. Cormen, C. E. Leiserson, R. Rivest, and C. Stein, *Introduction to Algorithms*, 2nd ed. McGraw Hill, 2001.

[42] M. Garey and D. Johnson, *Computers an Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.

[43] J. Diaz, M. D. Penrose, J. Petit, and M. Serna, “Convergence theorems for some layout measures on random lattice and random geometric graphs,” *Combinatorics, Probability and Computing*, vol. 9, pp. 489–511, 2000.

[44] C. Bettstetter, “On the minimum node degree and connectivity of a wireless multihop network,” in *MOBIHOC*, 2002.

[45] ——, “On the connectivity of wireless multihop networks with homogeneous and inhomogeneous range assignment,” in *Proceedings of Vehicular Technology Conference*, vol. 13, Vancouver, Canada, September 2002, pp. 1706–1710.

[46] C. Bettstetter and J. Zangl, “How to achieve a connected ad hoc network with homogeneous range assignment: An analytical study with consideration of border effects,” in *Proceedings of MWCN*, 2002, pp. 125–129.

[47] C. Bettstetter and C. Hartmann, “Connectivity of wireless multihop networks in a shadow fading environment,” *Wireless Networks*, vol. 11, no. 5, pp. 571–579, 2005.

[48] R. Hekmat and P. Van Mieghem, “Connectivity in wireless ad-hoc networks with a log-normal radio model,” *Mobile Networks and Applications*, vol. 11, no. 3, pp. 351–360, 2006.

[49] T. Rappaport, *Wireless communications: principles and practice*. Prentice Hall PTR Upper Saddle River, NJ, USA, 2001.

[50] K. Kobayashi, *Mathematics of information and coding*. Amer Mathematical Society, 2002.

[51] S. Weinzierl, “Introduction to monte carlo methods.” [Online]. Available: http://arxiv.org/abs/hep-ph/0006269v1

[52] E. W. Weisstein, “Circle-circle intersection,” from Mathworld – A Wolfram Web Resource. [Online]. Available: http://mathworld.wolfram.com/Circle-CircleIntersection.html

[53] T. Cormen, C. Leiserson, R. Rivest, and C. Stein, *Introduction to Algorithms*. Prentice Hall, 2004.

[54] D. Patterson, G. Gibson, and R. Katz, “A case for redundant arrays of inexpensive disks (RAID),” in *Proceedings of ACM SIGMOD International conference on Management of data*, 1988, pp. 109–116.

[55] Q. Malluhi and W. Johnston, “Coding for high availability of a distributed-parallel storage system,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 9, no. 12, pp. 1237–1252, 1998.

[56] A. Dimakis, V. Prabhakaran, and K. Ramchandran, “Decentralized erasure codes for distributed networked storage,” *IEEE/ACM Transactions on Networking (TON)*, vol. 14, no. SI, pp. 2809–2816, 2006.

[57] A. Jiang and J. Bruck, “Diversity coloring for distributed storage in mobile networks,” Tech. Rep., 2001.

[58] A. Dimakis, K. Ramchandran, Y. Wu, and C. Suh, “A Survey on Network Codes for Distributed Storage,” *Proceedings of the IEEE*, vol. 99, no. 3, 2011.

[59] S. Pawar, S. El Rouayheb, and K. Ramchandran, “On secure distributed data storage under repair dynamics,” in *Information Theory Proceedings (ISIT), 2010 IEEE International Symposium on*, 2010, pp. 2543–2547.

[60] Level 3 Communications, Network Map. [Online]. Available: http://www.level3.com/Resource-Library/Maps/Level-3-Network-Map.aspx

[61] S. El Rouayheb and K. Ramchandran, “Fractional repetition codes for repair in distributed storage systems,” in *Communication, Control, and Computing (Allerton), 2010 48th Annual Allerton Conference on*. IEEE, 2010, pp. 1510–1517.

[62] A. Jiang, M. Cook, and J. Bruck, “Optimal t-interleaving on tori,” in *Proceedings. International Symposium on Information Theory, 2004. ISIT 2004.*, 2004, p. 22.

[63] M. Sardari, R. Restrepo, F. Fekri, and E. Soljanin, “Memory allocation in distributed storage networks,” in *Proceedings IEEE International Symposium on Information Theory (ISIT), 2010*, 2010, pp. 1958–1962.

[64] M. Naor and R. Roth, “Optimal File Sharing in Distributed Networks,” *SIAM Journal on Computing*, vol. 24, pp. 158–183, 1995.

[65] A. Jiang and J. Bruck, “Network file storage with graceful performance degradation,” *ACM Transactions on Storage (TOS)*, vol. 1, no. 2, pp. 171–189, 2005.

[66] ——, “Memory allocation in information storage networks,” in *Proceedings. IEEE International Symposium on Information Theory, 2003.*, 2003, p. 453.

[67] ——, “Diversity coloring for distributed storage in mobile networks,” Tech. Rep., 2001.

[68] V. Vazirani, *Approximation algorithms*. Springer Verlag, 2001.

[69] C. Guo, G. Lu, D. Li, and et. al., “BCube: A High Performance, Server-centric Network Architecture for Modular Data Centers,” *ACM SIGCOMM*, 2009.

[70] A. Greenberg, J. Hamilton, N. Jain, and et. al., “VL2: A Scalable and Flexible Data Center Network,” *ACM SIGCOMM*, 2009.

[71] B. Heller, S. Seetharaman, P. Mahadevan, and et. al., “ElasticTree: Saving Energy in Data Center Networks,” *Networked Systems Design and Implementation*, 2010.

[72] Amazon EC2 Virtual Machines & Pricing. [Online]. Available: http://aws.amazon.com/ec2/pricing/

[73] David Chappell. Introducing Windows Azure. [Online]. Available: http://www.davidchappell.com/writing/white_papers/Introducing_the_Windows_Azure_Platform,_v1.4--Chappell.pdf

[74] A. Ganapathi, Y. Chen, A. Fox, R. Katz, and D. Patterson, “Statistics-Driven Workload Modeling for the Cloud,” *5th International Workshop on Self-Managing Database Systems*, 2011.

[75] D. Gmach, J. Rolia, L. Cherkasova, and A. Kemper, “Workload analysis and demand prediction of enterprise data center applications,” in *Proceedings of the 2007 IEEE 10th International Symposium on Workload Characterization*, 2007.

[76] A. Ganapathi, Y. Chen, A. Fox, R. H. Katz, and D. A. Patterson, “Statistics-driven workload modeling for the cloud,” *IEEE Internet Computing*, 2010.

[77] X. Y. Wang, D. jun Lan, X. Fang, M. Ye, and Y. Chen, “A resource management framework for multi-tier service delivery in autonomic virtualized environments,” in *IEEE/IFIP Network Operations and Management Symposium*, 2008.

[78] Y. Song, H. Wang, Y. Li, B. Feng, and Y. Sun, “Multi-tiered on-demand resource scheduling for vm-based data center,” in *Cluster Computing and the Grid*, 2009.

[79] B. Urgaonkar, P. Shenoy, A. Ch, and P. Goyal, “Dynamic provisioning of multi-tier internet applications,” in *In Proceedings of the 2nd International Conference on Autonomic Computing*, 2005.

[80] R. Urgaonkar, U. Kozat, K. Igarashi, and M. Neely, “Dynamic resource allocation and power management in virtualized data centers,” in *IEEE/IFIP Network Operations and Management Symposium*, 2010.

[81] J. Moore, J. Chase, P. Ranganathan, and R. Sharma, “Making scheduling "cool": temperature-aware workload placement in data centers,” in *Proceedings of the annual conference on USENIX Annual Technical Conference*, 2005.

[82] Q. Tang, S. Gupta, and G. Varsamopoulos, “Energy-efficient thermal-aware task scheduling for homogeneous high-performance computing data centers: A cyber-physical approach,” *IEEE Transactions on Parallel and Distributed Systems*, Nov 2008.

[83] H. Fraysseix, J. Pach, and R. Pollack, “How to draw a planar graph on a grid,” *Combinatorica*, vol. 10, no. 1, pp. 41–51, 1990.

APPENDIX A

A NP-COMPLETENESS PROOF OF *RBCDN-RP* AND *RBLCS-AP*

The *RBCDN-RP* and *RBLCS-AP* can be proved to be NP-complete by a transformation from the Hamiltonian Cycle in Planar Graph Problem (HCPGP) which is known to be NP-complete [42]. A *Hamiltonian Cycle* in an undirected graph $G = (V, E)$ is a simple cycle that includes all the nodes. A graph is a *planar* if it can be embedded in a plane by mapping each node to a unique point in the plane and each edge is a line connecting its endpoints, so that no two lines meet except at a common endpoint [42].

Hamiltonian Cycle in Planar Graph Problem (HCPGP)
INSTANCE: Given an undirected planar graph $G = (V, E)$.

QUESTION: Does G contains a Hamiltonian Cycle?

Theorem 10. RBCDN-RP *is NP-complete.*

Proof. It is easy to verify whether a set of additional edges of total cost $\leq C$ reduces the *RBCDN* of graph G with region R from $\alpha_R(G)$ to $\alpha_R(G) - K$. Therefore *RBCDN-RP* is in NP.

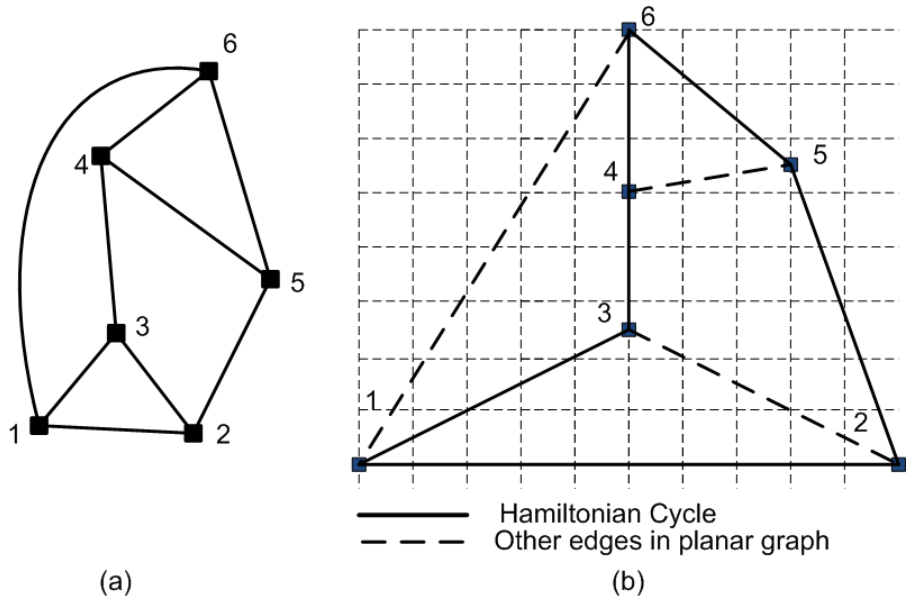


Figure 7.1: Transformation of a *HCPGP* instance to a *RBCDN-RP* and *RBLCS-AP* instance

From an instance of the *HCPGP* (a planar graph $G = (V, E)$) an instance of the *RBCDN-RP* (the layout $LG' = (P', L')$ of a graph $G' = (V', E')$) is created in the following way. First, a straight line embedding of the planar graph G is created on a plane so that lines corresponding to links in G do not intersect each other. Such an embedding can be carried out in polynomial time [83]. Let us call this layout $LG'' = (P'', L'')$. Another layout $LG' = (P'', \emptyset)$ is created, by setting $P' = P''$ and $L' = \emptyset$. The graph G' corresponding to the layout $LG' = (P', L')$ is

the instance of *RBCDN-RP* created from the instance of *HCPGP*. A region R is defined to be circular area of sufficiently small radius r, such that if a region fails, it can only destroy (i) a single node with all links incident on it, or (ii) a single link. Since the created instance of *RBCDN-RP* has no links ($E' = L' = \emptyset$), the *RBCDN* of G' with region R is n where $n = |V'| = |P'|$. The parameters C and K of the instance of the *RBCDN-RP* are set to be equal to n and $n-1$ respectively. The costs to the links of $\bar{E}'$ are assigned in the following way. The cost of a link $c(e)$ = 1, if $e \in (E \cap \bar{E}')$ and $c(e) = \infty$, if $e \in (\bar{E} \cap \bar{E}')$.

If the instance of the *HCPGP* has a Hamiltonian Cycle, we can use the set of links that make up the cycle, to augment the link set E' of the instance $G' = (V', E')$ of the *RBCDN-RP*. The augmented G', (G'_{aug}), is now a simple cycle that involves all the nodes. With the given definition of region R (a small circle of radius r), only one node can be destroyed when a region fails. Accordingly *RBCDN* of G'_{aug} is 1. It may be recalled that *RBCDN* of G' is n. Accordingly, augmentation of the link set of G' reduced its *RBCDN* by $n-1$. Due to the specific cost assignment rule of the links, the total cost of link augmentation is n. Therefore, if the *HCPGP* instance has a Hamiltonian Cycle, the *RBCDN* of the instance of *RBCDN-RP* can be reduced by K with augmentation cost $\leq C$.

Suppose that it is possible to reduce the *RBCDN* of the instance of *RBCDN-RP* by K with augmentation cost being at most C. This implies that the *RBCDN* of G' can be reduced from n to 1 (as $K = n-1$) when it is augmented with additional links with total cost at most n (as $C = n$). In order for the *RBCDN* of G'_{aug} to be 1, the node connectivity of G'_{aug} must be at least 2. A n node graph that has the fewest number of links and yet is 2-connected, is a cycle that includes all the nodes. As G' had no links, this implies at least n links must have been added to create the augmented graph G'_{aug}. Given that the cost of a link $c(e)$ = 1, if $e \in (E \cap \bar{E}')$ and $c(e) = \infty$, if $e \in (\bar{E} \cap \bar{E}')$, and total cost of link augmentation is at most n, it is clear that the links used in augmenting G must be from the set $(E \cap \bar{E}')$. These links are part of the edge set of the instance of *HCPGP*. Accordingly, the instance of *HCPGP* must have a Hamiltonian Cycle. □

Theorem 11. RBLCS-AP *is NP-complete.*

Proof. It is easy to verify whether a set of additional edges of total cost $\leq C$ increases the *RBLCS* of graph G with region R from $\gamma_R(G)$ to $\gamma_R(G) + K$. Therefore *RBLCS-AP* is in NP.

From an instance of the *HCPGP* (a planar graph $G = (V, E)$) an instance of the *RBLCS-AP* is created in the same manner as described in above proof. The *RBLCS* of G' with region R is 1. The parameters C and K of the instance of the *RBLCS-AP* are set to be equal to n and $n-2$ respectively. The costs to

the links of $\bar{E}'$ are assigned in the following way. The cost of a link $c(e)$ = 1, if $e \in (E \cap \bar{E}')$ and $c(e) = \infty$, if $e \in (\bar{E} \cap \bar{E}')$. Then the rest of the proof follows the same same proof for Theorem 7. □

APPENDIX B

NP-COMPLETENESS PROOF OF *DCRA* PROBLEM

The decision version of the *DCRA* problem can be shown to be NP-complete by a transformation from the *Subset Sum problem* which is known to be NP-Complete [42]. Decision version of em DCRA problem includes an additional parameter K and asks the following question: Is there a subset $R' \subseteq R$, such that the resource requirement of all requests in R' can be satisfied and the revenue $P(R')$ generated by R' is at least as large as K?

Problem 9 (Subset Sum Problem)**.** INSTANCE*: Given*
A set of elements U, a weight $w(u_l) \in \mathbb{Z}^+, \forall u_l \in U$, and a weight constraint W.

Question*: Is there a subset $U' \in U$ such that $\sum_{u_l \in U'} w(u_l) = W$?*

Theorem 12. *DCRA problem is NP-complete.*

Proof. It is easy to see that DCRA is in NP since a nondeterministic algorithm needs only to guess a subset R' of the requests and check in polynomial time if both the VM and fault domain constraints are satisfied by R' and revenue $P(R')$ generated by R' is at least the objective value K.

Subset Sum problem can be transformed to DCRA problem in polynomial time. Given an instance of Subset Sum problem, an instance of DCRA problem is created. The instance of DCRA problem has only one server P and only one type of virtual machine T_1. P hosts W VMs $\{v_1, v_2, \ldots, v_W\}$ of type T_1. There is only one fault domain F containing P. In other words, fault domain F contains VMs $\{v_1, v_2, \ldots, v_W\}$. Corresponding to every element $u_l \in U$, in the instance of the subset sum problem, a request R_l given by $< T_1, w_l, 1 >$ is created, where w_l corresponds to the weight $w(u_l)$ of element $u_l \in U$. So the total set of requests R is of same size as U. The unit cost of VM of type T_1 is 1 and the fixed cost associated with a fault domain is 0. The objective value K is set to be W.

The instance of DCRA problem created using the above construction rule will have a set of satisfiable requests $R' \subseteq R$ such that $P(R')$ is at least K if and only if the instance of Subset Sum problem has a subset $U' \in U$ such that $\sum_{u_l \in U'} w(u_l) = W$.

Suppose that the set U in the instance of subset sum problem contains a subset U' such that $\sum_{u_l \in U'} w(u_l) = W$. Then, in the instance of DCRA problem, only the subset of requests $R_l \in R'$, corresponding to each element $u_l \in U'$, need to be satisfied. The revenue generated by these requests, in that case, will be equal to $\sum_{R_l \in R'} w_l = \sum_{u_l \in U'} w(u_l) = W = K$. So a solution of subset sum problem instance will give a feasible solution of DCRA instance.

On the other hand, suppose there is a set of requests $R' \subseteq R$ that can be satisfied and the revenue generated is equal to K ($K = W$ is the maximum number of VMs present in the instance). Then a subset U' of U can be created such that U' contains only those elements u_l which corresponds to a request $R_l \in R'$. In that case, the sum of the subset U' is given as $\sum_{u_l \in U'} w(u_l) = \sum_{R_l \in R'} w_l = K = W$. So, a feasible solution of DCRA problem gives a feasible solution of subset sum problem. This completes the proof. □

BIOGRAPHICAL SKETCH

Sujogya Banerjee graduated from the School of Computing, Informatics and Decision Systems Engineering at Arizona State University. He received his Bachelor's degree in Computer Science and Engineering from Jadavpur University, India, in 2007. He started his Ph.D. in Computer Science under supervision of Dr. Arunabha Sen later that year. His research interests include combinatorial optimization techniques, fault tolerant design and analysis of networks, wireless networks, resource allocation in data storage networks and data center networks.

His research focuses on network resource allocation and efficient network design techniques, in presence of spatially correlated faults or region-based faults, in both wired and wireless domain. He introduced several novel network fault-tolerant metrics appropriate for wired and wireless networks under region-based failures and utilizing these metrics he proposed methods of design and analysis of networks. Also, he presented efficient resource allocation techniques in data storage networks and data center networks.

He also has shown considerable amount of interest in solving optimization problems in optical networks and in solving graph partition problems related to social networks.

Below is a list of his publications:

Conferences:

1 "On Region-based Fault Tolerant Design of Distributed File Storage in Networks", Sujogya Banerjee, Shahrzad Shirazipourazad and Arunabha Sen, In Proceedings of IEEE International Conference on Computer Communications (INFOCOM)(Mini-Conference), Orlando, USA, March 2012

2 "On a Fault-tolerant Resource Allocation Scheme for Revenue Maximization in Data Centers", Sujogya Banerjee, Sudheendra Murthy and Arunabha Sen, In Proceedings of IEEE Conference on Advanced Networks and Telecommunication Systems (ANTS), Bengaluru (Bangalore), India, December 2011

3 "Beyond Connectivity - New Metrics to Evaluate Robustness of Networks", Sujogya Banerjee, Shahrzad Shirazipourazad, Pavel Ghosh and Arunabha Sen, In Proceedings of IEEE Conference on High Performance Switching and Routing (HPSR), Cartagena, Spain, July 2011

4 "Design and Analysis of Networks with Large Components in Presence of Region-Based Faults", Sujogya Banerjee, Shahrzad Shirazipourazad and Arunabha Sen, In Proceedings of IEEE International Conference on Communication (ICC), Kyoto, Japan, June 2011

5 “Impact of Region-Based Faults on the Connectivity of Wireless Networks in Log-normal Shadow Fading Model”, Sujogya Banerjee and Arunabha Sen, In Proceedings of IEEE International Conference on Communication (ICC), Kyoto, Japan, June 2011

6 “Impact of Region-based Faults on the Connectivity of Wireless Networks”, Arunabha Sen, Sujogya Banerjee, Pavel Ghosh and Shahrzad Shirazipourazad In Proceedings of Allerton Conference on Communication, Control, and Computing, Urbana-Champaign, October 2009

7 “Region-Based Connectivity - A New Paradigm for Design of Fault-tolerant Networks”, Arunabha Sen, Sudheendra Murthy and Sujogya Banerjee In Proceedings of IEEE Conference on High Performance Switching and Routing (HPSR), Paris, France, June 2009

Journals:

1 “Region-Based Connectivity : A New Evaluation Metric of Fault-tolerant Wireless Networks”, Sujogya Banerjee, Arunabha Sen, Bao Hong Shen, Ling Zhou, Bin Hao and Sudheendra Murthy, Submitted for review in IEEE Transactions of Networking

2 “Design of Distributed Data Storage Networks Robust Against Region-Based Faults”, Sujogya Banerjee, Shahrzad Shirazipourazad and Arunabha Sen, Submitted for review in IEEE Transactions of Networking

Other Publications:

1 “Perspective Analysis for Online Debates”, Sukru Tikves, Sedat Gokalp, Mhamed Temkit, Sujogya Banerjee , Jieping Ye and Hasan Davulcu, In Proceedings of International Symposium on Foundation of Open Source Intelligence and Security Informatics (FOSINT-SI), Istanbul, Turkey, August 2012

2 “A System for Ranking Organizations Using Social Scale Analysis” (extended Journal Version), Sukru Tikves, Sujogya Banerjee, Hamy Temkit, Sedat Gokalp, Hasan Davulcu, Arunaba Sen, Steven Corman, Mark Woodward, Shreejay Nair, Inayah Rochmaniyah, and Ali Amin, In Journal of Social Network Analysis and Mining

3 “Partitioning Signed Bipartite Graphs for Classification of Individuals and Organizations”, Sujogya Banerjee, Kaushik Sarkar, Sedat Gokalp, Arunabha Sen and Hasan Davulcu, In Proceedings of International Conference on Social Computing, Behavioral-Cultural Modeling and Prediction (SBP), College Park, Maryland, USA, April 2012

4 “A System for Ranking Organizations Using Social Scale Analysis”, Sukru Tikves, Sujogya Banerjee, Hamy Temkit, Sedat Gokalp, Hasan Davulcu, Arunaba Sen, Steven Corman, Mark Woodward, Inayah Rochmaniyah, and Ali Amin, In Proceedings of International Symposium on Open Source Intelligence & Web Mining (OSINT-WM), Athens, Greece, September 2011

5 “Dynamic Lightpath Allocation in Translucent WDM Optical Networks”, Subir Bandyopadhyay, Quazi Rahman (University of Windsor), Sujogya Banerjee, Sudheendra Murthy, Arunabha Sen (Arizona State University), In Proceedings of IEEE International Conference on Communication (ICC), Dresden, Germany, June 2009

6 “Brief announcement: On Regenerator Placement Problems in Optical Networks”, Arunabha Sen, Sujogya Banerjee, Pavel Ghosh, Sudheendra Murthy (Arizona State University) and Hung Ngo (University of Buffalo (SUNY)), In Proceedings of 22nd ACM Symposium on Parallelism in Algorithms and Architectures (SPAA) Santorini, Greece, June 2010