2004

Survivable multicasting in WDM optical networks

Sateesh Chandra Shekhar
Louisiana State University and Agricultural and Mechanical College

SURVIVABLE MULTICASTING
IN
WDM OPTICAL NETWORKS

A Thesis

Submitted to the Graduate Faculty of the
Louisiana State University and
Agricultural and Mechanical College
in partial fulfillment of the
requirements for the degree of
Master of Science of Electrical Engineering

in

The Department of Electrical Engineering

By
Sateesh Chandra Shekhar
Bachelor of Engineering (Computer Engineering)
University of Pune, 1999
August 2004

Dedicated to The Lord. He who smiles while we discover his manifestations.

Acknowledgements

I would like to sincerely express my gratitude to Dr. Ahmed El-Amawy for his invaluable guidance and mentorship in my current research project. He was deeply instrumental in setting the direction of this research work. I am also grateful to him for allowing me to work at my own pace within his expectations. Finally, I truly appreciate his gesture of meeting me on holidays and at times other than his office hours.

As I look back on one and half years that I have spent at LSU, I thank my parents, Maj. and Mrs. K. Balasubramanian, for all their support and patience while I have been in school. My educational experience here has been possible only because of their interest in and positive encouragement of my desire to pursue higher studies.

In addition, I thank Dr. Sai Pinnepalli at the Office of Facility Services at LSU, who supported me financially and allowed me the flexibility of an excellent work and study environment during my entire stay at LSU.

Finally, I thank Stephan Pascu for the invaluable brainstorming that we frequently indulged in, in meeting rooms, corridors, and near water fountains in the ECE Department. Those discussions deeply impacted the shape of my research.

Table of Contents

List of Figures

Abstract

Opportunities abound in the global content delivery service market and it is here that multicasting is proving to be a powerful feature. In WDM networks, *optical splitting* is widely used to achieve multicasting. It removes the complications of optical-electronic-optical conversions [1]. Several multicasting algorithms have been proposed in the literature for building *light trees*. As the amount of fiber deployment increases in networks, the risk of losing large volumes of data traffic due to a fiber span cut or due to node failure also increases. In this thesis we propose heuristic schemes to make the primary multicast trees *resilient* to network impairments. We consider single link failures only, as they are the most common cause of service disruptions. Thus our heuristics make the primary multicast session *survivable* against single link failures by offering alternate multicast trees.

We propose three algorithms for recovering from the failures with *proactive* methodologies and two algorithms for recovering from failures by *reactive* methodologies. We introduce the new and novel concept of *critical subtree*. Through our new approach the proactive and reactive approaches can be amalgamated together using a criticality threshold to provide recovery to the primary multicast tree. By varying the criticality *threshold* we can control the amount of protection and reaction that will be used for recovery.

The performance of these five algorithms is studied in combinations and in standalone modes. The input multicast trees to all of these recovery heuristics come from a previous work on designing power efficient multicast algorithms for WDM optical networks [1]. Measurement of the power levels at receiving nodes is indeed indicative of

the power efficiency of these recovery algorithms. Other parameters that are considered for the evaluation of the algorithms are network usage efficiency, (number of links used by the backup paths) and the computation time for calculating these backup paths. This work is the first to propose metrics for evaluating recovery algorithms for multicasting in WDM optical networks. It is also the first to introduce the concept of hybrid proactive and reactive approach and to propose a simple technique for achieving the proper mix.

Chapter 1

WDM Optical Networks

Fiber optic communications have provided us with high-speed communications with enormous bandwidth potential. Although fibers can support very high data rates (nearly 50 terabits per second, [6]), the associated electronic processing hardware will typically not be able to keep up with such speeds. Hence, electronic handling of data at network nodes basically limits the throughput of the network. Further, electronic processing is required because optical storage and processing technologies are not mature yet. Hence, data that must be stored or processed at an intermediate node has to be converted to its electronic form and stored in an electronic buffer memory. A routing decision is then made and the data is then queued at the output port, converted back into its optical form and transmitted towards its final destination.

Networks can be classified into three generations depending on the technology used at the physical level. The first generation networks used copper based technologies. Second generation networks used a combination of copper and optical technologies. Third generation networks are all optical networks. These networks are yet to become practical because of the challenges involved in routing and buffering in the optical domain without an intermediate conversion to the electronic domain.

Current networks use a mix of copper and optical based technologies. To improve the throughput of the network and to minimize transmission delay the network architecture must both reduce the number of times a message is processed by the intermediate nodes (and thus reduce the number of times an optical signal is converted back and forth between the electronic and optical domains) and must streamline the

processing at each node. The irregular nature of most existing networks doesn't necessarily allow this. Hence complex routing tables are often used to make routing decisions less complicated and less time-consuming. If the network could be connected in a regular uniform pattern, routing decisions could be significantly simplified thereby reducing the processing times at the intermediate nodes [30]. But because of many real world constraints, a regular uniform pattern in building a network may not be feasible. Also economic reasons necessitate reuse of existing fiber connections in networks which restricts the physical topology options [30].

1.1 Optical Fiber Principles

According to [7], light has an information carrying capacity 10,000 times greater than the highest radio frequencies. Advantages of optical fibers over copper transmission lines include the ability to carry signals over long distances with low error rates, immunity to electrical interference, and security [7, 8]. The first fiber optic communication had been experimentally tested in the nineteenth century. However, it was in the second half of the twentieth century that the technology began to advance rapidly and began being used in practical networks. After the viability of transmitting light over fiber had been established, the next step in the development of fiber optics was to find a light source that would be sufficiently powerful and narrow. *Light emitting diodes* (LED) and *laser diodes* (LD) proved capable of meeting these requirements. Researchers in the mid 1960s proposed that optical fiber might be a suitable transmission medium for light. There was an obstacle, however, and that was the loss of signal strength (or attenuation) in the glass with which they were working. In 1970, Corning produced the first communication-grade *fibers* [7]. With attenuation less than 20 decibels

per kilometer (dB/km), this purified glass fiber exceeded the threshold for making fiber optics a viable technology [8].

A basic optical communication system shown in Figure 1.1 consists of an optical *transmitter*, an optical *receiver* and optical fiber as the communication medium. There are several other components that go into the system to make it practical, such as optical add/drop multiplexers, optical amplifiers, switches and wavelength converters.

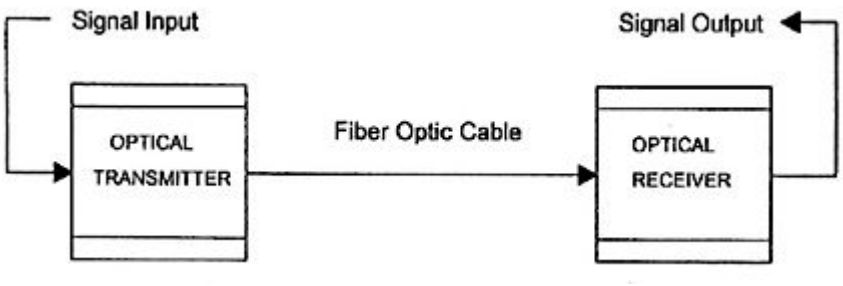

Figure 1.1 Optical Transmission System

A fiber optic cable is composed of two concentric layers known as the *core* and the *cladding*. Figure 1.2 illustrates these two layers. The core and cladding have different

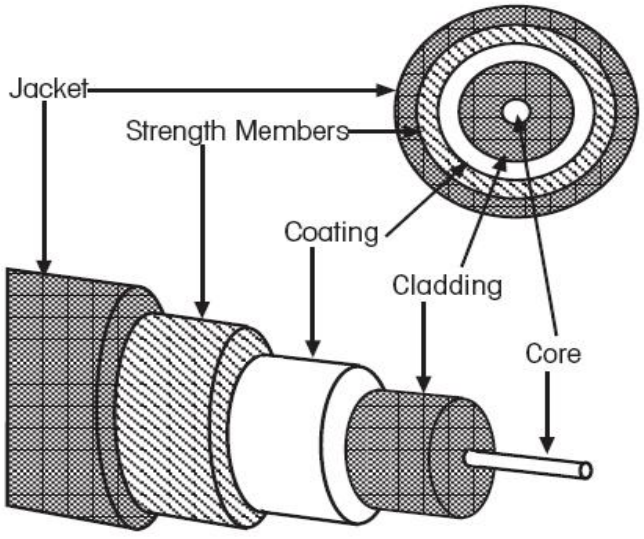

Figure 1.2 Core, Cladding, and Coating

indices of refraction with the core having n_1 and the cladding n_2 ($n_1 < n_2$) [11]. Controlling the angle at which the lightwaves are transmitted makes it possible to control how efficiently they reach their destination. Lightwaves are guided through the core of the optical fiber in much the same way that radio frequency (RF) signals are guided through coaxial cable. The lightwaves are guided to the other end of the fiber by being reflected within the core. The composition of the cladding glass relative to the core glass determines the fiber's ability to reflect light. Having a higher *refractive index* in the core of the glass than in the surrounding cladding glass creates the required "*waveguide*." The refractive index of the core is increased by slightly modifying the composition of the core glass, generally, by adding small amounts of a *dopant*. Alternatively, the waveguide can be created by reducing the refractive index of the cladding using different dopants. The fiber optic cable has an additional protective coating around the cladding known as the jacket. The jacket usually consists of one or more layers of polymer. It protects the core and cladding from shocks that might affect their optical or physical properties.

There are two general categories of optical fiber: *single-mode* and *multimode* (see Figure 1.3).

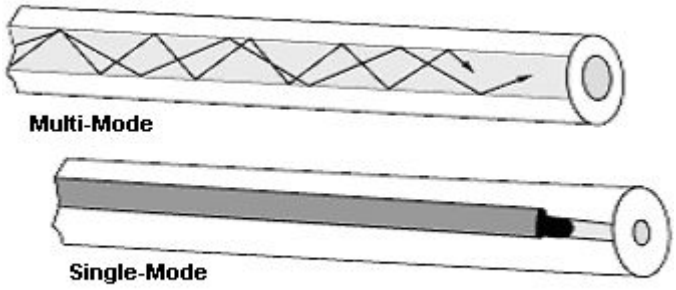

Figure 1.3 Single and Multimode Fiber

Multimode fiber was the first type of fiber to be commercialized. It has much larger core than single-mode fiber, allowing hundreds of modes of light to propagate through the fiber simultaneously. Additionally, the larger core diameter of multimode fiber facilitates the use of lower-cost optical transmitters (such as light emitting diodes or vertical cavity surface emitting lasers) and connectors. Single-mode fiber, on the other hand, has a much smaller core that allows only one mode of light at a time to propagate through the core [1]. While it might appear that multimode fibers have higher capacity, in fact the opposite is true. Single-mode fibers are designed to maintain *spatial* and *spectral integrity* of each optical signal over longer distances, allowing more information to be transmitted. Its tremendous information-carrying capacity and low intrinsic loss have made single-mode fiber the ideal transmission medium for a multitude of applications. Single-mode fiber is typically used for longer-distance and higher-bandwidth applications (see Figure 1.4) [19].

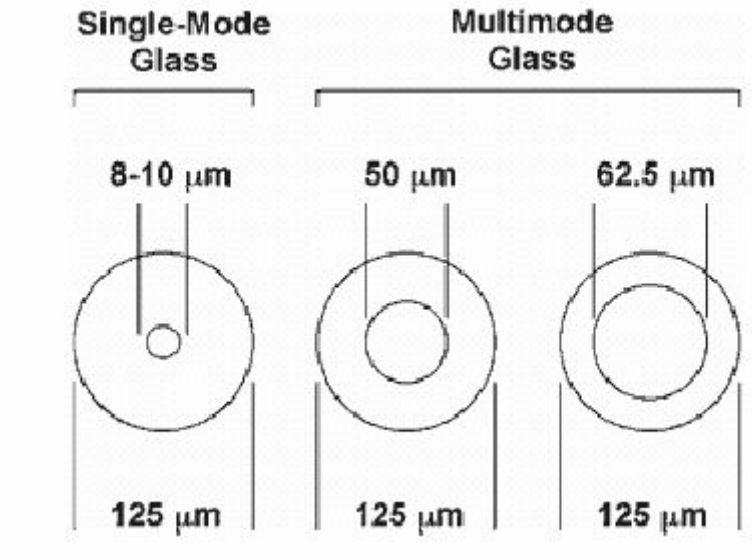

Figure 1.4 Optical Fiber Sizes

Multimode fiber is used primarily in systems with short transmission distances (under 2 km), such as premises communications, private data networks and parallel optic applications.

1.2 Wavelength Division Multiplexing

WDM enables the utilization of a significant portion of the available fiber bandwidth by allowing many independent signals to be transmitted simultaneously on one fiber, with each signal being on a different wavelength ([13]). Routing and detection of these signals can be accomplished independently, with the wavelength determining the communication path by acting as the signature address of the origin, destination or routing. Components are therefore required that are wavelength selective, allowing for the transmission, recovery, or routing of specific wavelengths.

In a simple WDM system, shown in Figure 1.5 each laser must emit light at a different wavelength with all the lasers' light multiplexed together onto a single optical fiber.

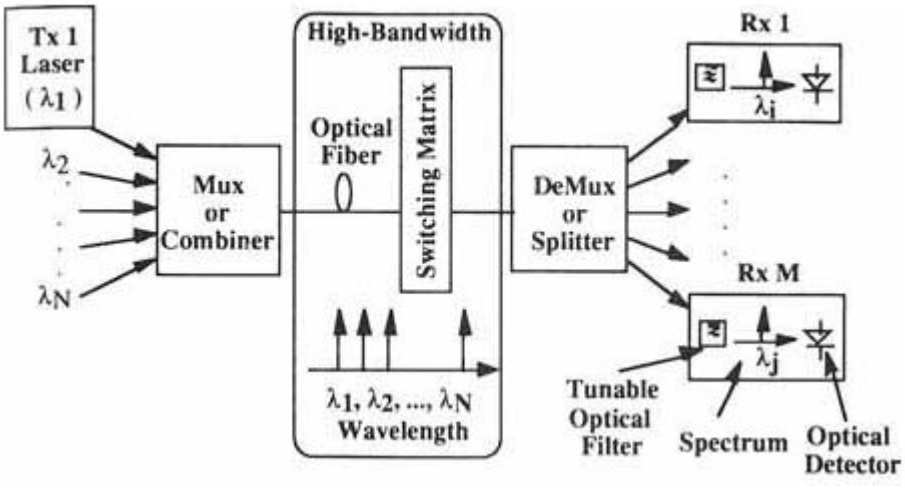

Figure 1.5 A Simple WDM System

After being transmitted through a high-bandwidth optical fiber, the combined optical signals must be de-multiplexed at the receiving end. One way to do that is to distribute the total optical power to the output ports and then require that each receiver selectively recovers only one wavelength using a tunable optical filter. Each laser is modulated at a given speed and the total aggregate capacity being transmitted along the high-bandwidth fiber is the sum total of the bit rates of the individual lasers.

1.3 Components of WDM System

The building blocks of a WDM system are briefly described in this section.

1.3.1 Optical Amplifiers

In fiber optic communications systems, problems arise from the fact that no fiber material is perfectly transparent. The visible-light or infrared (IR) beams carried by a fiber are attenuated as they travel through the material. This necessitates the use of *repeaters* in spans of optical fiber longer than about 100 kilometers.

A conventional repeater puts a modulated optical signal through three stages: (1) optical-to-electronic conversion, (2) electronic signal amplification, and (3) electronic-to-optical conversion. Repeaters of this type limit the bandwidth of the signals that can be transmitted in long spans of fiber optic cable. This is because, even if a laser beam can transmit several gigabits per second of data, the electronic circuits of a conventional repeater cannot.

The commercial development of WDM networks was made possible by the development of *optical amplifiers* known as *EDFA's* (*Erbium Doped Fiber Amplifiers*) which provide a way to optically amplify all the wavelengths at the same time, regardless of their individual bit rates, modulation schemes or power levels [16]. An EDFA

amplifier is an optical repeater that amplifies a modulated laser beam directly, without opto-electronic and electro-optical conversion. The device uses a short length of optical fiber doped with the rare earth element erbium. When the signal-carrying laser beam pass through this fiber, external energy is applied, usually at infrared wavelengths. This so-called pumping excites the atoms in the erbium-doped section of optical fiber increasing the intensity of the laser beams passing through. The beams emerging from the EDFA retain all of their original modulation characteristics, but are higher in energy than the input beams.

1.3.2 Add/Drop Multiplexer

In many WDM networks it is necessary to drop some traffic at intermediate points along the route between the end points. A wavelength add/drop multiplexer (WADM) is used for that purpose. A typical WADM is shown in Figure 1.6.

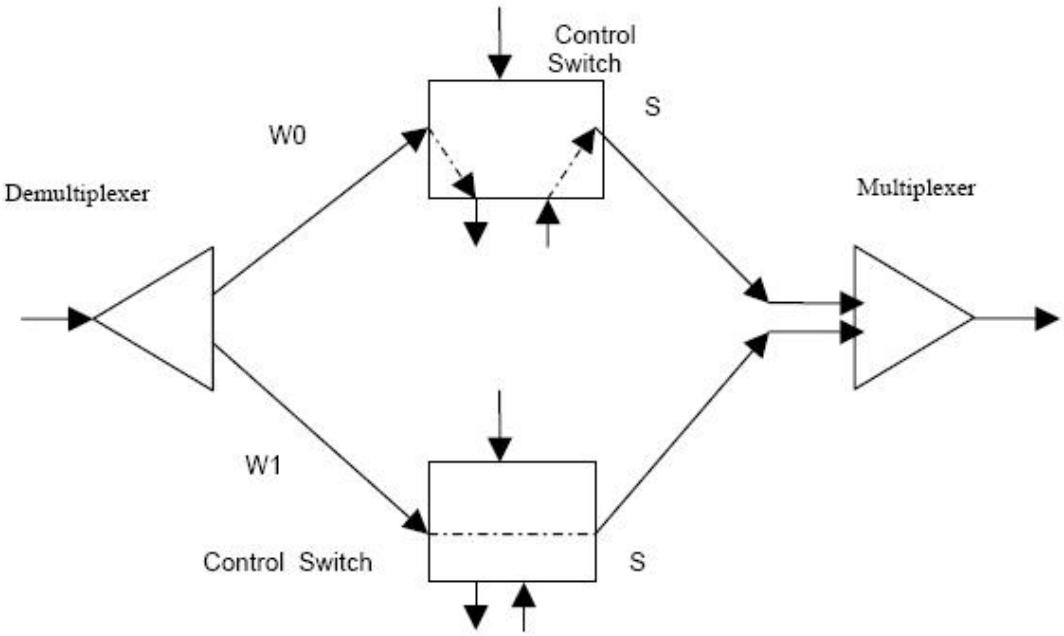

Figure 1.6 Wavelength Add/Drop Multiplexer

A WADM can be realized using 2x2 switches and a de-multiplexer. If the control switch is in the *bar state*, then the signal on the corresponding wavelength passes through the WADM. If the switch is in the *cross state*, then the signal on the corresponding wavelength is dropped locally, and another signal of the same wavelength may be added [6].

1.3.3 Wavelength Crossconnect

Efficient use of fiber facilities at the optical level obviously becomes critical as service providers begin to move wavelengths around the world. In the optical domain, a network element is needed that can accept various wavelengths on input ports and route them to appropriate output ports in the network. *Routing* and *grooming* are key areas that must be addressed. This is the function of the OXC, as shown in Figure 1.7.

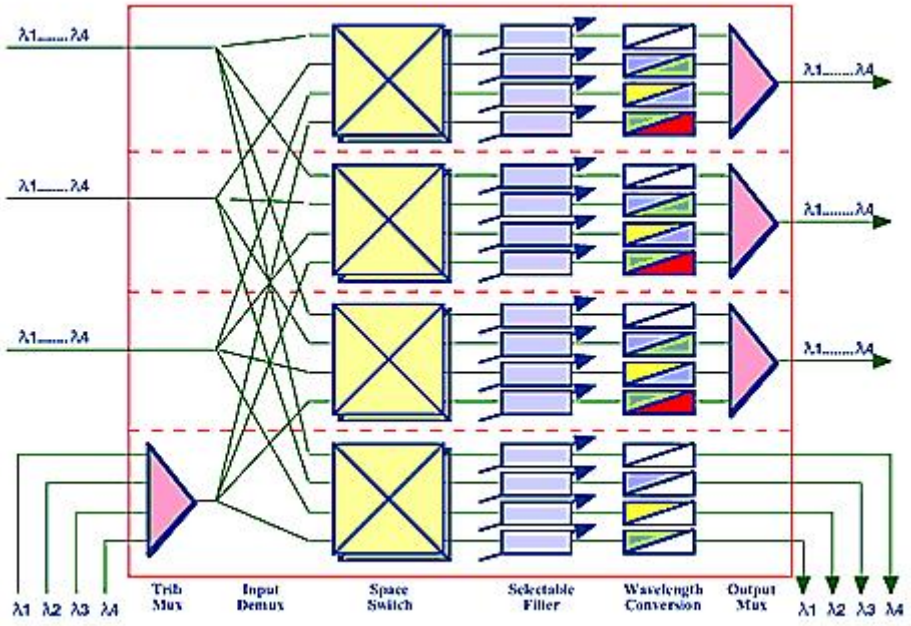

Figure 1.7 Wavelength Crossconnect Block Diagram

The function of this element is to provide (under network control), the ability to connect or switch any input wavelength channel from an input fiber (or port) to any one of the output fibers (or ports) in the optical domain [18]. Digital cross-connect systems are deployed en masse and provide the critical function of grooming traffic to fill output ports on the system efficiently. To accomplish this, the OXC needs three building blocks (See Figure 1.8):

Fiber switching - the ability to route all of the wavelengths on an incoming fiber to a different outgoing fiber.

Wavelength switching - the ability to switch specific wavelengths from an incoming fiber to multiple outgoing fibers.

Wavelength conversion - the ability to take incoming wavelengths and convert them (on the fly) to another optical frequency on the outgoing port. This may be necessary to achieve strictly non-blocking architectures when using wavelength switching.

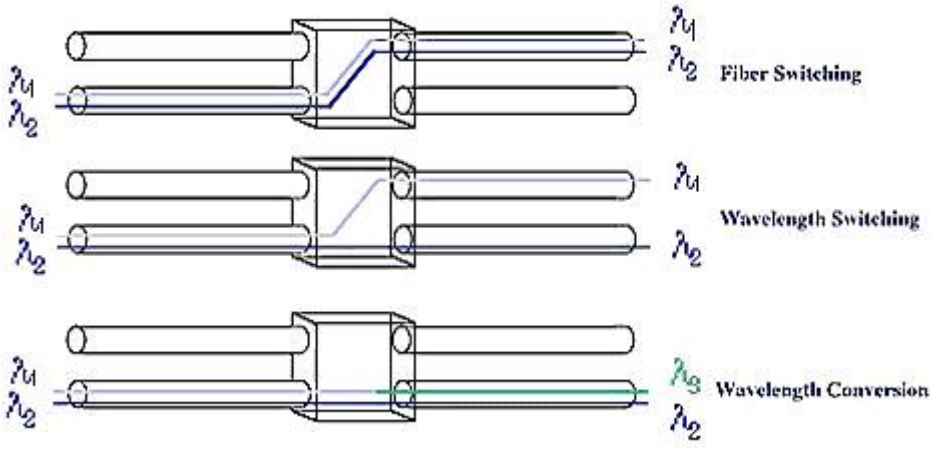

Figure 1.8 Optical Cross-Connects

1.4 WDM Optical Network Architectures

The following section describes the three major classes of WDM optical network architectures, which are, broadcast and select networks, wavelength routed networks, and linear lightwave networks.

1.4.1 Broadcast and Select Networks

In this type of network (see Figure 1.9) all the nodes are connected to a central *star coupler*, which mediates all the communications among the nodes in the network. All the nodes in the network are equipped with fixed number of tunable transmitters and receivers. To receive a particular wavelength all the destinations tune their receivers to that wavelength and start receiving the signal. All simultaneous transmissions occur at different wavelengths.

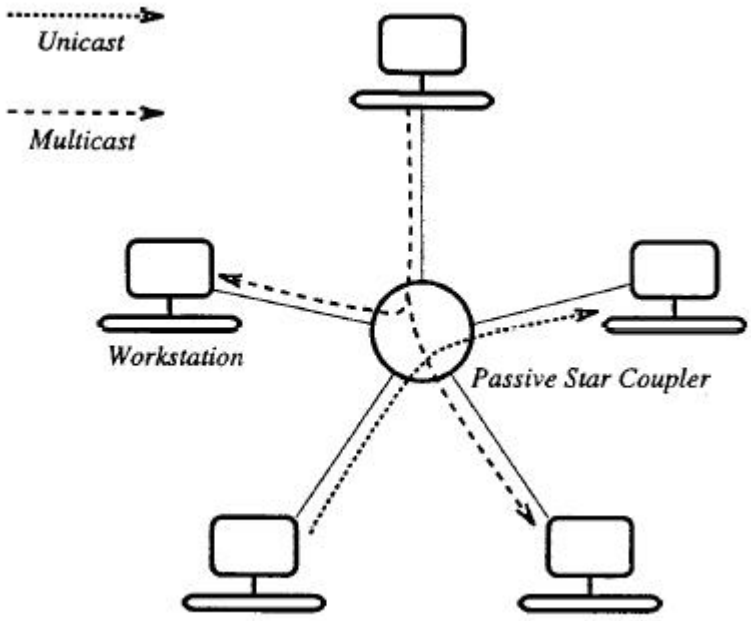

Figure 1.9 Broadcast and Select Network

The role of the star coupler is to combine all these signals and then to broadcast the combined signal to all the nodes. Multiple communications can take place concurrently by appropriately tuning the receivers. These networks involve single hop transmission to the destination without any intermediate opto-electronic conversion. The problem with this type of network is that of collision which occurs when two or more nodes try to transmit simultaneously on the same wavelength. Also, power is not efficiently utilized. Several protocols have been proposed for scheduling communications in such networks [20], [21], and [22].

1.4.2 Wavelength Routed Networks

A wavelength routed network consists of *Wavelength Cross Connects* interconnected by point-to-point fiber links in an arbitrary topology. Each end-user is connected to an active switch via a fiber link. Each node consists of transmitters and receivers, both of which may be wavelength tunable. See Figure 1.10 for illustration.

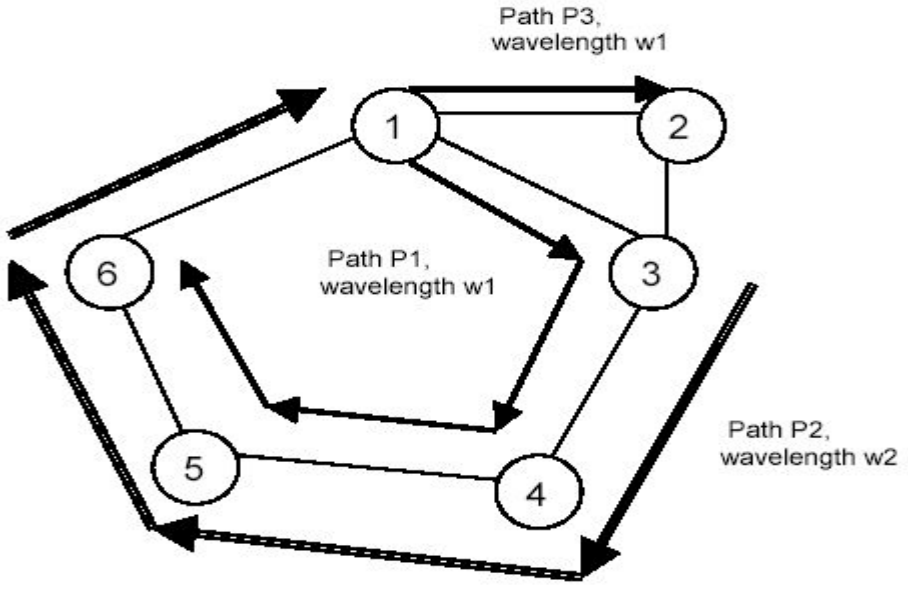

Figure 1.10 Wavelength Continuity Constraint and Wavelength Reuse

The end-nodes tune their transmitters and receivers to the wavelength used for the *lightpath*. A lightpath is an all-optical communication channel between two nodes in the network and may span more than one fiber link. The basic requirement in a wavelength-routed optical network is that no two lightpaths traversing the same fiber link can use the same wavelength channel and that a lightpath uses the same wavelength across all links that it traverses. See Figure 1.10 for illustration. Selecting routes and wavelength is referred to as the routing and wavelength assignment (RWA) problem. This problem has been heavily studied [6]. The following techniques can be used for routing:

Fixed routing: Only one route is used for any given node pair. This path is usually the shortest path

Alternate routing: Two or more routes are provided for each given node pair. These routes are searched one by one in a pre-determined order. (usually increasing order of hop lengths).

Exhaust routing: All possible routes are searched for each given node pair and the most suitable one is selected. This routing produces the best result but is computationally complex.

The wavelength assignment is also usually done using one of several schemes. A sample of such schemes is described below:

Most used: This method gives preference to the wavelength, which is used, on the largest number of links in the network. The wavelengths are searched in decreasing order of use.

Least used: In this method wavelengths are searched in non-decreasing order of their utilization in the network and allocated accordingly.

Random: This method randomly selects a wavelength from the available wavelengths.

Round Robin: This method assigns wavelengths in a round robin fashion from the pool of available wavelengths [6].

All the selected lightpaths may not utilize shortest paths. Some shortest paths may have to be sacrificed in order to maximize the number of allowable lightpaths. Thus, one may allow several alternate routes for lightpaths to be established.

The wavelength continuity constraint causes bandwidth losses. One efficient way to overcome this problem is to use wavelength converters. A wavelength converter is an optical device, which is capable of shifting a signal on one wavelength to another wavelength [6]. A wavelength converter of degree D can shift an incoming wavelength to any of the D possible wavelengths at that converter.

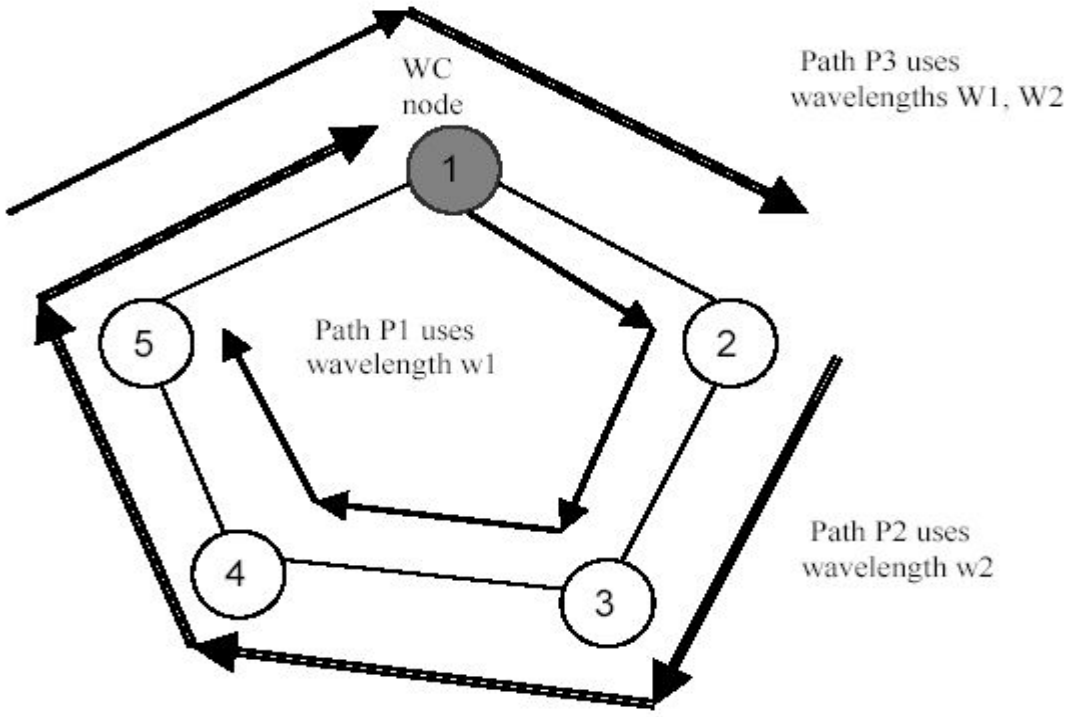

Figure 1.11 Wavelength Routing with Waveband Converters

If the conversion degree equals the number of wavelengths on the fiber, then the converter is said to have full conversion capability. If its degree is less than the number of

wavelengths on the fiber, then it is a partial wavelength converter. Nodes with wavelength conversion capability are known as *wavelength conversion* (WC) nodes. Nodes that do not have this capability are known as *wavelength interchange* (WI) nodes. A wavelength convertible network performs better than a WI network as the wavelength converters relax the wavelength continuity constraint thereby giving better bandwidth utilization. Figure 1.11 illustrates this fact. In the network shown, if there are only two wavelengths, the path P3 between nodes 5 and 2 cannot be established as the wavelengths are locked in the existing paths P1 and P2. However if there is a wavelength converter at node 1 then path P3 can be realized. Wavelength W1 is used for the link (5, 1) and the signal is shifted to wavelength W2 at node 1 thereby using W2 for the link (1, 2). So path P3 uses two wavelengths. It must be stated that wavelength converters are not yet practical [6].

1.4.3 Linear Lightwave Networks

These networks utilize the idea of partitioning the usable optical spectrum into wavelengths or *wavebands* as shown in Figure 1.12. These networks use two levels of *partitioning* and several such wavebands are multiplexed on a fiber. Several wavelengths are multiplexed onto a single waveband. So, unlike a wavelength routed network, linear lightwave network nodes de-multiplex, switch, and multiplex wavebands not wavelengths. Since the linear lightwave network doesn't distinguish between individual wavelengths within a waveband individual wavelengths are separated from each other at the receiving node.

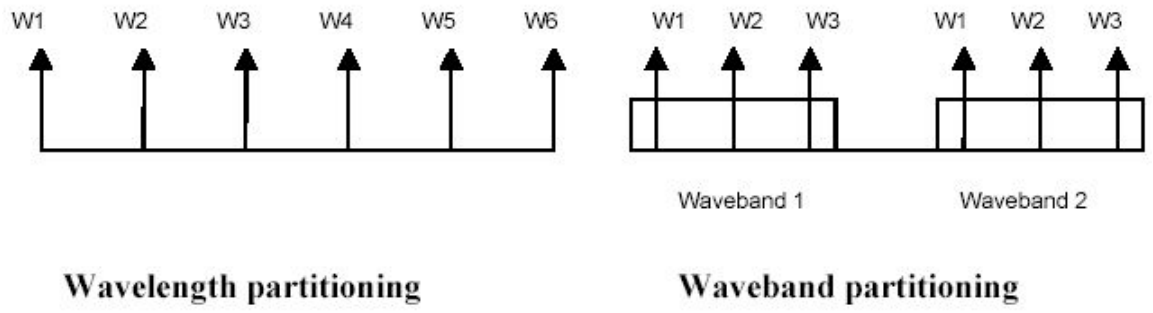

Figure 1.12 Wavelength and Waveband Partitioning

These networks have the additional constraints of *inseparability* and *combining* signals from distinct sources. According to the inseparability constraint, channels belonging to the same waveband when combined on the same fiber cannot be separated within the network. Thus they travel together after the point where they were combined. The distinct source combining constraint states that on any fiber only signals from distinct sources are allowed to be combined.

1.5 Future of WDM Optical Networks

Optical networks provide easy to manage high bandwidth services both for Internet exchanges and for local area networking applications. The introduction of WDM technology in optical fiber networks can be considered as a way of replacing the central switching functions to distributed network functions like optical add/drop multiplexers or repeaters [23]. Thus, WDM supported Optical Internet and related services are expected to be a major driving force in future networking architectures. It is predicted that it will contribute towards substantially reducing the complexity and thus the cost of future Internet services. There is a great potential to eventually move to an all optical switching and all optical routing architectures as these technologies mature.

Chapter 2

Multicasting and Recovery Strategies: A Review

2.1 Multicasting

WDM based network infrastructures provide high capacity and cost effective transport networks. The advantage of high speed, high network connectivity and high bandwidth should be complimented with *resilience* to network failures. *Multicasting* refers to a communication pattern in which information from a single source is sent to multiple destinations. In a *point-to-point* network, a transmission by a node is received only by the node at the other end of the link [1], [26]. However in a single channel (wavelength) multicast network transmission is received by all the nodes attached to the channel [1]. On any channel of a WDM network at any given point of time, a transmission by a node is transmitted to all the nodes listening to that channel. Multicasting can be implemented on either single hop networks or multi-hop networks. In single hop networks, a star coupler is used to mediate the communications between all the nodes. This form of communication renders the single hop networks practically unsuitable for larger distances. It is here that the more practical multi-hop networks come into play

Multicasting in optical networks can be generally achieved in three different ways [21].

1. Making multiple copies of the same message electronically and transmitting a copy to each destination. However, this method would need many optical-electronic–optical conversions at every node which makes this method inefficient.

2. Multiple *unicasts* is another method for multicasting. In this case we separately route the message from the source to each destination. This method however, is not very appropriate as huge bandwidth is required. That bandwidth grows with the multicast group size (number of destinations in a multicast session).
3. Making multiple copies of the same message optically. This is by far the best method since it maximizes the bandwidth savings. This method is also known as light splitting.

The first two methods are not taken into account in WDM networks. Supporting multicast in WDM networks by the third method requires the optical switches at the nodes to have the *light splitting* (or copying) capability in order to be able to forward multiple copies of data in the optical domain [26]. A WDM network may be a *sparse splitting network* [31], where only a subset of the switches support light splitting. To achieve multicasting on such a sparse splitting WDM network, several algorithms like *Re-route-to-Source*, *Re-route-to-Any*, *Member-First,* and *Member-Only* have been proposed in the literature [26].

The fault model considered in this thesis assumes that at any time only one link can fail. A link failure can cause tremendous loss of information, affecting potentially a huge number of network users. That is why network survivability is an important issue. The algorithms developed in this thesis use some of the ideas put forth in the above mentioned algorithms for multicasting in sparse split WDM networks. The concepts of Re-route-to-Any and Re-route-to-Source are central to our recovery algorithms as will be explained in Chapter 3.

2.1.1 Light Trees

Ideally a multicast session in a WDM optical network will be implemented using a *light tree*. A light tree is a tree rooted at the multicast source and includes all the destinations [1]. The advantage of light trees is that a single wavelength can be used for any link in the tree. So the entire tree uses a single wavelength. This approach helps significantly in facilitating concurrent multicast sessions. In some cases it may not be possible to service all destinations with a single multicast tree. In such a case it is necessary to construct a *multicast forest* employing one wavelength per tree.

2.1.2 Light Splitting

Light splitting or power splitting is a concept, which is unique to the optical medium. A light splitter is a passive device used to distribute input signal to a set of out going edges of the node. A node that is capable of splitting the signal is called a multicast capable (MC) node. Not all nodes in the network will have this capability. Some nodes cannot split light. Those are called multicast incapable (MI) nodes. Building an MC node is more expensive compared to a building an MI node. So, all nodes in the network need not be MC nodes. Since not all the nodes in the network have splitting capability, it may not be possible to reach all destinations through a single light tree. It may be required to utilize more than one light tree. A *collection* of such trees is known as a *light forest* [26] or *multicast forest*. MC and MI nodes usually employ a Splitter-and-Delivery Cross Connect or a Tap-and-Continue Cross Connect respectively for their purposes.

2.1.3 Splitter-and-Delivery Cross-Connect

The SAD is an MC cross-connect that was proposed in [32]. An NxN cross-connect consists of a set of SAD switches for each wavelength. An NxN SAD switch consists of an interconnection of N power splitters, N^2 optical gates (to reduce the excessive crosstalk), and N^2 X 1 photonic switches [32]. Figure 2.1 shows the organization of a cross-connect based on the SAD switch. A SAD switch with a splitting capacity of S need not always split S times. Depending on the tree it can split M=1 or 2 or S times. A large number of splitters in the network will have the negative effect of the need for optical amplifiers and high fabrication cost.

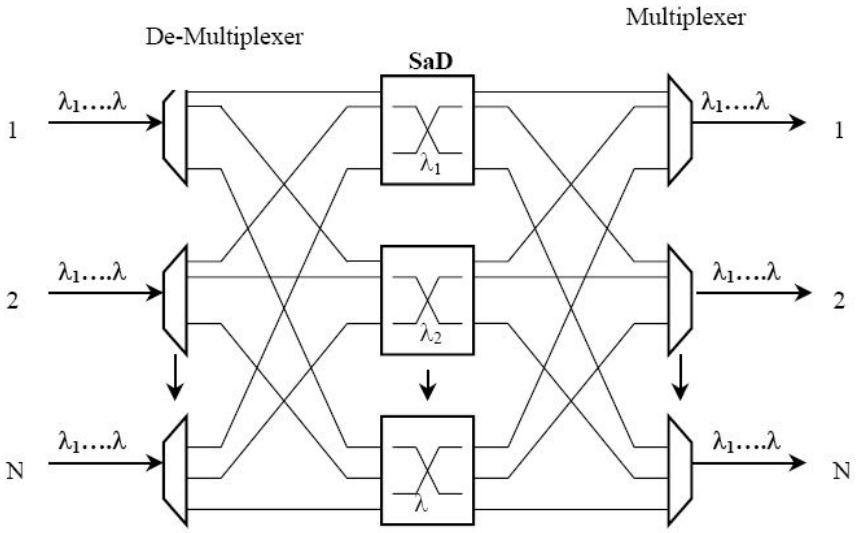

Figure 2.1 An NXN SAD Cross Connect

2.1.4 Tap-and-Continue Cross-Connect

Figure 2.2 shows an implementation of a TAC cross-connect. A TAC is a typical node which cannot split. It can just send a small part of the signal power to the local node and forward the remaining signal to the successor(s) of the local node. An NxN TAC cross-connect supporting W wavelengths uses a set of W 1xN Tap-and-Continue

Modules (TCMs). In TCM`s, a small fraction of the input signal is tapped and forwarded to the local station. The remaining power which is more than 90% is switched to any one of the other outputs.

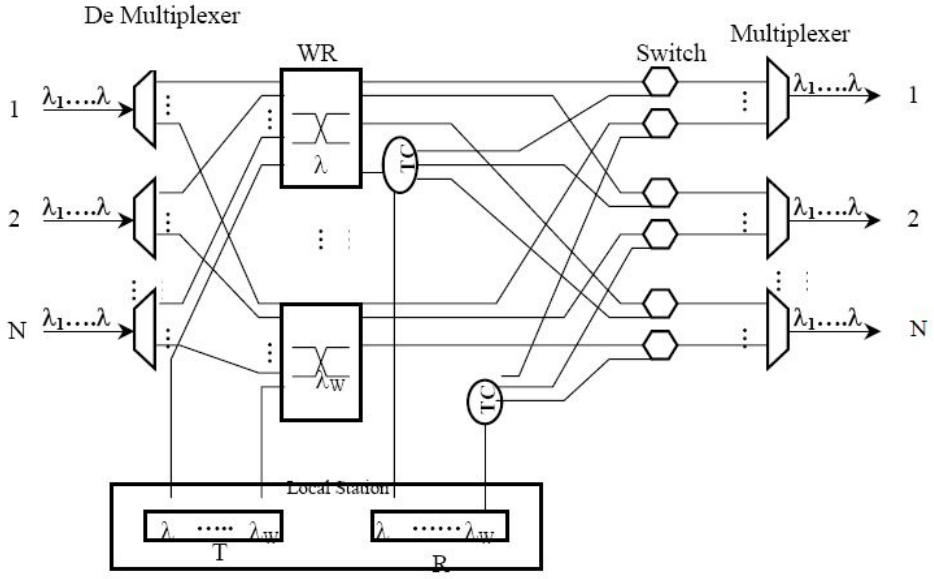

Figure 2.2 An NXN TAC Cross Connect

2.1.5 Power Losses

The problem of power losses was not encountered in electronic packet or circuit switched networks and thus was not addressed in the existing multicast routing algorithms for such networks. In order to maintain reliability in optical networks the power (or signal strength) received by any node in the tree must be above a certain fixed level [5]. This level is mandated by the ability of the light detectors at the destination to recover the information reliably [1]. The losses that an optical signal suffers as it travels from source to destination are basically of two types-losses due to optical splitting and losses due to signal attenuation. An *M*-way splitter splits a signal of power *P* into *M* signals of power *P/M*. In reality the power of each successor of a splitting node will be lower than P/*M* due to some multiplexing losses. Every time the signal travels some

distance, be it through splitters or without splitters it always undergoes attenuation losses, which depend largely on the distance traveled by the signal.

2.2 Recovery Strategies

Many failure *recovery* mechanisms have been proposed for generic networks in the literature. The application of these mechanisms is not restricted to optical networks alone. Classifications have been done based on the nature of route computation, (i.e. centralized or distributed), by the layer where they take place (WDM, MPLS, IP…), by the type of protection (link-based or path-based), and by the computation timing (pre-computed or real time). The works in [3] and [4] consider two main categories for classifying the recovery schemes at the broadest level, i.e. *proactive* and *reactive* techniques. The proactive or protection techniques allocate and reserve the backup resources in advance, thus, providing fast recovery on preplanned paths at the expense of

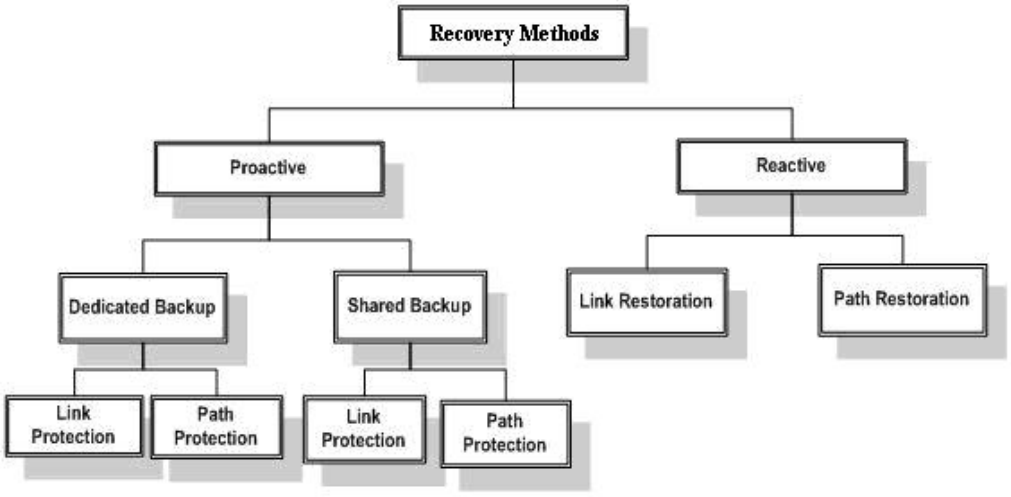

Figure 2.3 Classification of Restoration Methods

an inefficient use of resources. The reactive or *restoration* methods can be classified as illustrated in Figure 2.3.

The restoration techniques make use of real time availability of resources. They provide a slower recovery but they do not reserve the resources for backup paths. The latency of restoration schemes will thus be higher than that of protective schemes. It is noteworthy that the two techniques can *coexist* in the same network, as we shall describe later.

Broadly speaking the methods are classified as protective or reactive. In the reactive approach, when a failure occurs, a search for an alternate path is initiated. In the absence of failures or with a few failures, the overhead of the reactive approach is low. However, this approach may not be successful if there are no resources present at the time of actual recovery. In case the recovery is computed in a distributed fashion contention may occur to win over the resources that are being needed to recover from some other failure simultaneously. This will result in several retries to recover. In the proactive approach, the backup paths are computed and the resources are reserved at the time of establishing the primary session. This method reserves resources, is faster, and always guarantees restoration.

The proactive or reactive schemes can be either *link based* or *path based*. When a component fails, *link based methods* select an alternate path between the end nodes of the failed link. This alternate path along with the intact part of the primary path is used for the recovery. This method is illustrated in Figure 2.4, which shows a primary lightpath, p1, and two backup lightpaths, b11 and b12, on a wavelength. When link 0−1 fails, backup path b11 is used. It can be observed that b11 is routed around link 0−1 while retaining the working segment of p1.

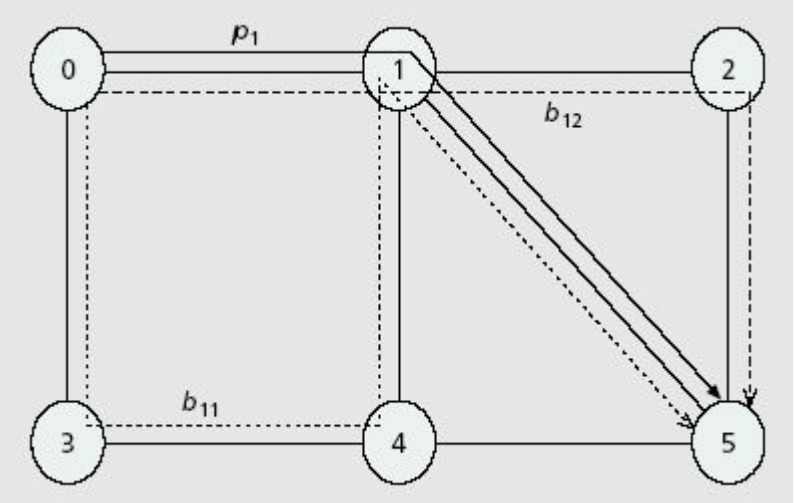

Figure 2.4 Link-Based Backup Path Reservation

When link 1−5 fails backup path b12 is used. It can be observed that b12 is routed around link 1−5 while retaining the working segment of p1. Note that the working segment of the primary lightpath is retained in the backup path.

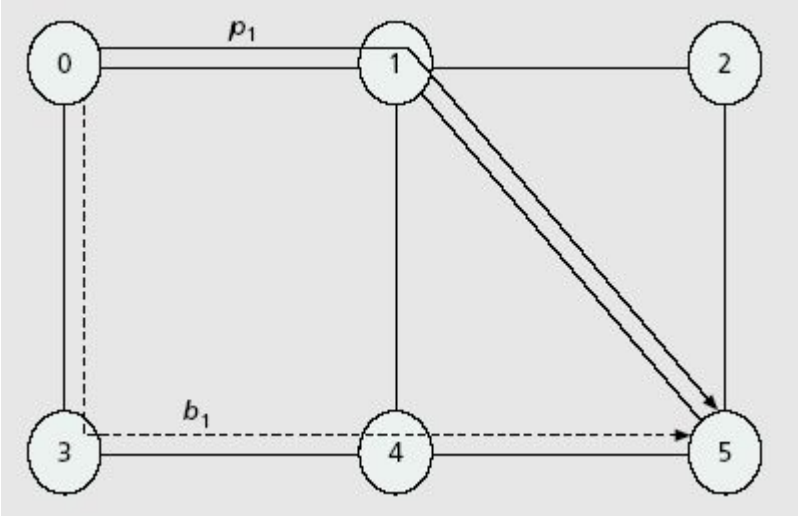

Figure 2.5 Path-Based Backup Path Reservation

In the case of *path based methods*, a backup path is computed between the end nodes of the failed primary lightpath. The backup path can use any wavelength independent of the one used by the corresponding primary lightpath. The path-based restoration method is illustrated in Figure 2.5. Figure 2.5 shows a primary lightpath, p1,

and its backup lightpath, b1, on a given wavelength. Note that b1 is established between the end nodes of p1, and the working segment of p1 is not utilized by b1.

If none of the channels are shared between any two backup channels, then the method is referred to as *dedicated backup restoration*. This method ends up reserving a lot of resources and is not resource efficient. This method is illustrated in Figure 2.6. The figure shows two primary lightpaths, p1 and p2, and their respective backup lightpaths b1 and b2. It can be observed that b1 and b2 do not share any wavelength channel.

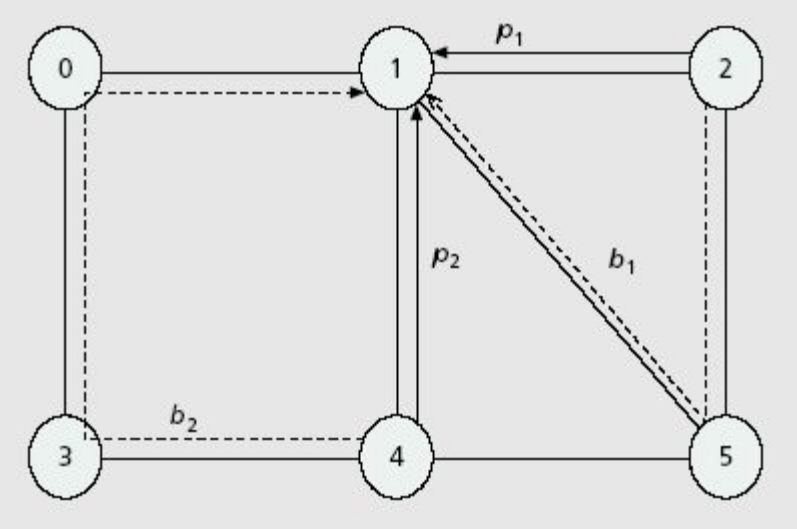

Figure 2.6 Dedicated Backup Path Reservation

If no two failures can occur at the same time then their backup channels can share channels. This is referred to as *shared backup restoration* and is illustrated in the Figure 2.7. The figure shows two primary lightpaths p_1 and p_2 and their respective backup lightpaths b_1 and b_2 on a certain wavelength. As p_1 and p_2 are disjoint, they do not fail at the same time under the single link failure fault model. Therefore, b_1 and b_2 can share the wavelength on link 5−1. This shared channel will be used by b_1 when link 2 − 1 fails and

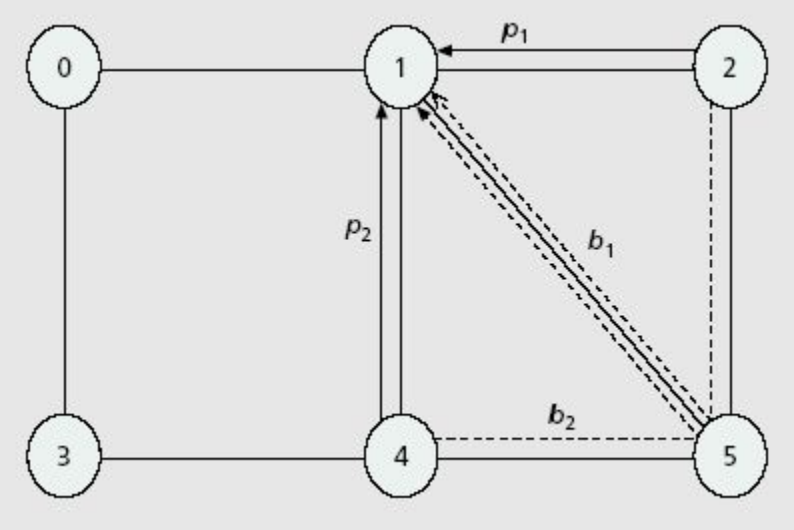

Figure 2.7 Shared Backup Path Restoration

by b_2 when link 5−1 fails.

It is to be noted that "*segment*" based recovery is also possible. In this case backup is provided at the segment level rather than the link or path level, where a segment is a subpath.

2.3 Strategies for Backup Path Computation

In this section we describe the strategy that has been proposed in [2] for finding backup trees. If the primary and backup trees are totally edge disjoint trees, sharing no edges in common, then an edge failure cannot affect both trees. Due to constraints on the connectedness of practical networks it might not be feasible to find another *link-disjoint* light tree to be used as a backup tree. For example, Figure 2.8 illustrates the case where it is not possible to find a backup tree which is edge disjoint from the primary tree. The primary paths are indicated in green color. Relaxing the condition to that of finding an arc disjoint tree we can find an alternate backup tree as illustrated in Figure 2.9. We employ the notion of arc-disjoint path which requires that the primary and backup paths not share

links in the same direction. This is also known as *directed-link disjointness*. The backup path is indicated in green dotted lines. We can see that the primary path and the backup path do not share the same direction on any link.

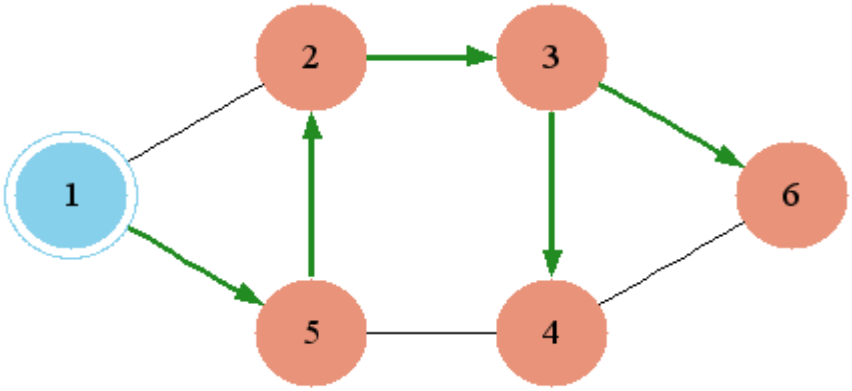

Figure 2.8 A Network with Multicast Session

For example, in order to protect the multicast session with source node 1, it is difficult to find a link disjoint tree to the destination nodes since the graph obtained after removing primary tree links is a disconnected network.

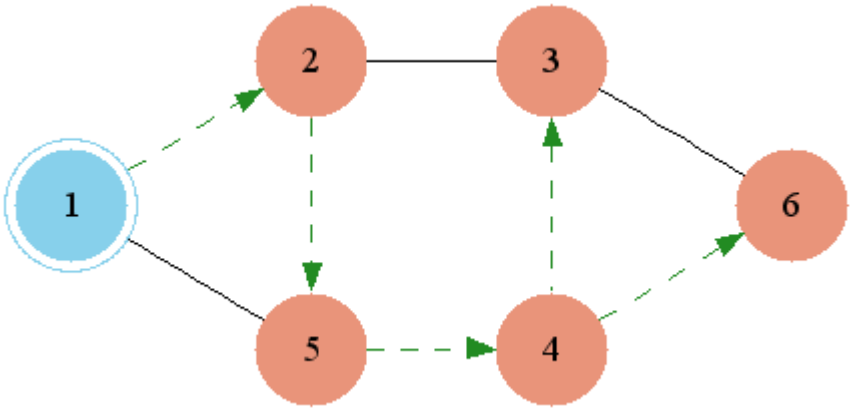

Figure 2.9 Arc Disjoint Tree

The idea of arc disjointedness is not new and has been explored in contexts other than optical networks. If a link is used bi-directionally, one for the primary path and the other for the backup path, then in the event of failure of the link we would treat as a failure of both the primary and backup paths on that link. We use this notion in our

heuristics for finding recovery paths in the next chapter. This notion is illustrated in Figure 2.10.

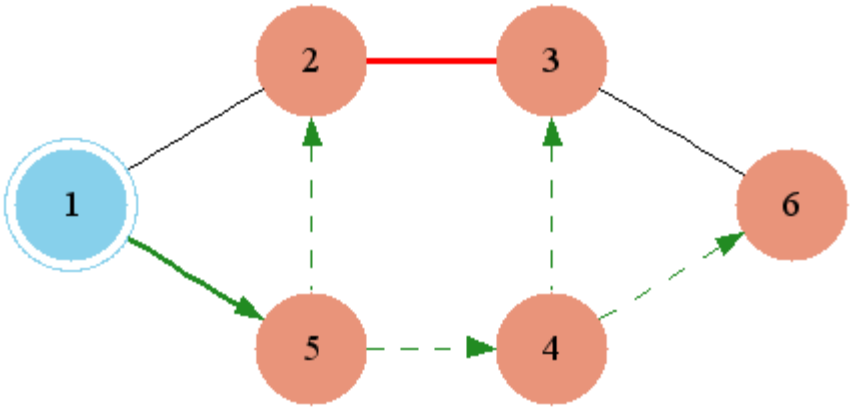

Figure 2.10 Failure Recovery

In our example if we were to fail the link between nodes 2 and 3 (indicated by a red edge), then we can recover from that failure using the tree illustrated in Figure 2.10. If we had to find a backup tree totally disjoint from the primary tree then in that case we could not have found one. Instead we use the concept of direction disjoint path to find such a tree. Since our primary aim is to just omit the failed link, (or set of links), our recovered sessions contain both primary path and the backup path. We can clearly see in Figure 2.10 that part of the primary tree is still intact and is being preserved in the recovered session. In addition we have some changes in the direction of transmission as indicated by the dotted lines. The bold green lines indicate the remnants of the primary tree and the dashed green lines correspond to the backup tree.

2.4 Power Efficient Multicast Algorithms

The input multicast sessions used to study the heuristics in this thesis come from the work in [1]. The algorithms in [1] attempt to reduce the degradation in the quality of the signal as it travels through the optical network. As mentioned earlier, there are two

dominant sources for *power losses* i.e. *attenuation losses* and *splitting losses*. There are essentially two choices, either to reduce splitting losses or to reduce attenuation losses. While building the tree, none of the losses should be allowed to attain significant levels on any branch of the tree. The heuristics in [1] employed certain techniques like backtracking and pruning to improve signal strengths at the destinations.

Backtracking was used for the first time in [1] as a novel approach to improve power delivery to the destinations. It can be treated as a first level of pruning. The second and third levels of pruning utilize balancing techniques to adjust and improve the power being delivered to the destinations. In our implementation of the heuristic algorithm to generate power conscious multicast trees, we adopted the *minimal* backtracking algorithm, (ALGORITHM 2 in [1]) and implemented a version without the second and third levels of pruning. ALGORITHM 2 in [1] is based on a shortest path algorithm and uses a mild version of the backtracking technique to construct the primary multicast tree. The output of this heuristic can be a single multicast tree or multiple multicast trees in a multicast forest.

Among the three heuristics proposed in [1] we chose Algorithm 2 over Algorithm 1 based on the results which showed that it performed better than Algorithm 1 in most of the cases in so far as power delivery is concerned. Algorithm 3 was ruled out from consideration because of its high time complexity.

2.5 Problem Definition

As the number of bandwidth hungry applications running on WDM optical networks grows, the significance of multicasting increases. Multicast sessions are based on light trees which utilize light splitting at various nodes. Even a single *fiber cut* on a

light tree based multicast session can disrupt the transmission of information to several destination nodes. It becomes all the more important to provide recovery options to these multicast sessions to avoid such disruptions. In this thesis we make these multicast trees survivable to single link failures.

The problem of establishing both primary and backup trees at a minimum cost such that the two trees do not share a link in the same direction is NP complete [2]. The complexity of the problem grows with the size of the network. The closest work addressing a problem similar to ours is reported in [2]. It deals with the problem for a sparse split networks through mathematical formulations for an ILP solver. In the treatment of the problem, the networks considered were small and the need for efficient heuristics that can address the problem for larger practical sized networks (with few hundred nodes) was acknowledged [2]. ILP formulations are practical for small-to-moderate-sized networks (a few tens of nodes). For a one-time static network design, it might be feasible to solve larger instances of the formulation given a longer solution time [2]. For larger networks, we need to have efficient heuristics that can generate good quality solutions within a reasonable time. Moreover, the work in [2] does not address the power efficiency of the recovery schemes. In this work we propose heuristic solutions to recover the multicast sessions for such large networks. In [2], a dedicated backup light tree is considered for each primary multicast session that is being protected. In our work we explore the possibility of resource sharing among backup paths, thus bringing down the overall cost of simultaneously protecting several multicast sessions.

Five heuristic algorithms will be introduced and discussed in the following chapters. The heuristics are classified into proactive and reactive schemes based on their

computation times (pre-computed or real time). The five heuristics approach the problem in distinct ways in order to attain the goal of ensuring resilience to single link failure.

As we mentioned earlier, proactive and reactive schemes can co-exist in a network. We propose a novel technique to blend the two classes of schemes in a hybrid recovery approach. Using this technique the amount of recovery to be done by the proactive and the reactive schemes can be adjusted thus providing part protection and part reaction-based recovery to the same primary multicast tree.

All five heuristics are studied for their performance against each other and also in combinations. The usage of network resources for providing recovery, average signal power being delivered upon recovery and computation time of the heuristics are studied in depth for comparison.

Chapter 3 describes the algorithms in detail with discussions and illustrations of recovering from the single link failures. Chapter 4 presents the results obtained on the standalone algorithms and the hybrid schemes. Chapter 5 describes the design of the software for the simulator. Chapter 6 is a conclusion.

Chapter 3

Survivable Multicasting

The objective of the recovery techniques is to make the primary multicast session resilient to network impairments. Our methods build on the power conscious multicast sessions generated by the power efficient multicast algorithms described by the work in [1]. While constructing the primary multicast trees in the multicast forest, we attempt to minimize the attenuation and splitting losses.

The previous work in [2], for survivable primary multicast sessions, brings out the need for efficient heuristics for large practical networks. This thesis attempts to develop heuristics that can solve the problem for large practical networks. There are essentially two classes of approaches for providing resilience to network failures: protective or reactive approaches. As we mentioned earlier, proactive and reactive schemes can *co-exist* in a network. We propose a novel approach to combine the proactive and reactive approaches in a *hybrid* treatment. Our new approach will give the designer the flexibility in deciding on the amount of protective or reactive based recovery provided. The value of a parameter called *threshold* determines the components that are to be provided with protective recovery. Any link whose failure prevents at least a number of destinations equal to “threshold” from receiving the signal is deemed *critical* and is a candidate for protective recovery. The rest of the links in the multicast session are *non-critical* and these are recovered using the reactive approaches. The simulator we developed (see Chapter 5) has been written to combine any of the protective approaches with any of the reactive approaches to provide recovery. The choice of the specific schemes in such a hybrid approach and the amount of recovery divided between the proactive and reactive

schemes can be made based on the simulation results obtained using different recovery algorithms. These results are discussed extensively in Chapter 4. Section 3.1 next describes the basics that will help the reader in understanding the heuristics. Section 3.2 introduces and explains three protection-based heuristics and the two reaction-based heuristics that we developed for making the primary multicast sessions survive single link failures. It is to be noted that any of our heuristics can be used in either a proactive or a reactive mode. But adhering to the general objective of fast reactive schemes, we have chosen only those schemes which compute the backup paths in less time for reaction-based schemes.

3.1 Basics

In the following we introduce and discuss some novel concepts that will be used in our heuristics. Such concepts allow us to control the balance between protective and reactive schemes in a unified framework.

3.1.1 Critical Tree

The critical tree is defined as the portion of the primary multicast tree in which failure of a component prevents *at least* a particular number of destinations (defined as *threshold)*, from receiving power. The Critical tree is directly related to the value of *threshold*, as described next.

3.1.2 Threshold

The threshold is defined as the number of destinations for which lack of power triggers the recovery using a reaction-based recovery mechanism. The remaining destinations form part of the critical tree which will be recovered using proactive mechanisms. Failure of a given link prevents destinations whose number is greater than

or equal to the threshold from receiving transmission, then that link becomes part of the critical tree. Once all such links are determined, one can determine the critical tree.

The threshold can be seen as a logical boundary in the treatment of the primary multicast tree. It divides the multicast tree into two portions, determined by the criticality of the nodes. One portion becomes the critical tree and is recovered using protection while the other portion is recovered using a reactive approach.

The approaches discussed here offer the option of controlling the amount of protection and restoration that will be used for recovery. By varying the value of threshold between 1 and the number of destination nodes in the primary multicast tree, we can go from providing complete protection to providing complete reaction based recovery to the primary session.

On one end, a threshold value of 1 indicates that the entire primary session is to be recovered using protection based schemes. On the other end, a threshold value equal to the number of destinations in the primary session indicates that the entire primary session is to be recovered using reaction based schemes. A value between these two extremes indicates choosing a mixed approach.

Let’s consider the following USA Longhaul network to illustrate the concept of threshold. A particular multicast session is shown in Figure 3.1.

Source of multicast session: 14

Multicast membership group: {10, 11, 12, 13, 16, 18, 20, 23, 24, 25, and 26}

Consider the case where the threshold is equal to 1. The nodes in dark salmon color are destinations while the source is indicated by a sky blue color. All the edges which are part of the critical tree are shown in red.

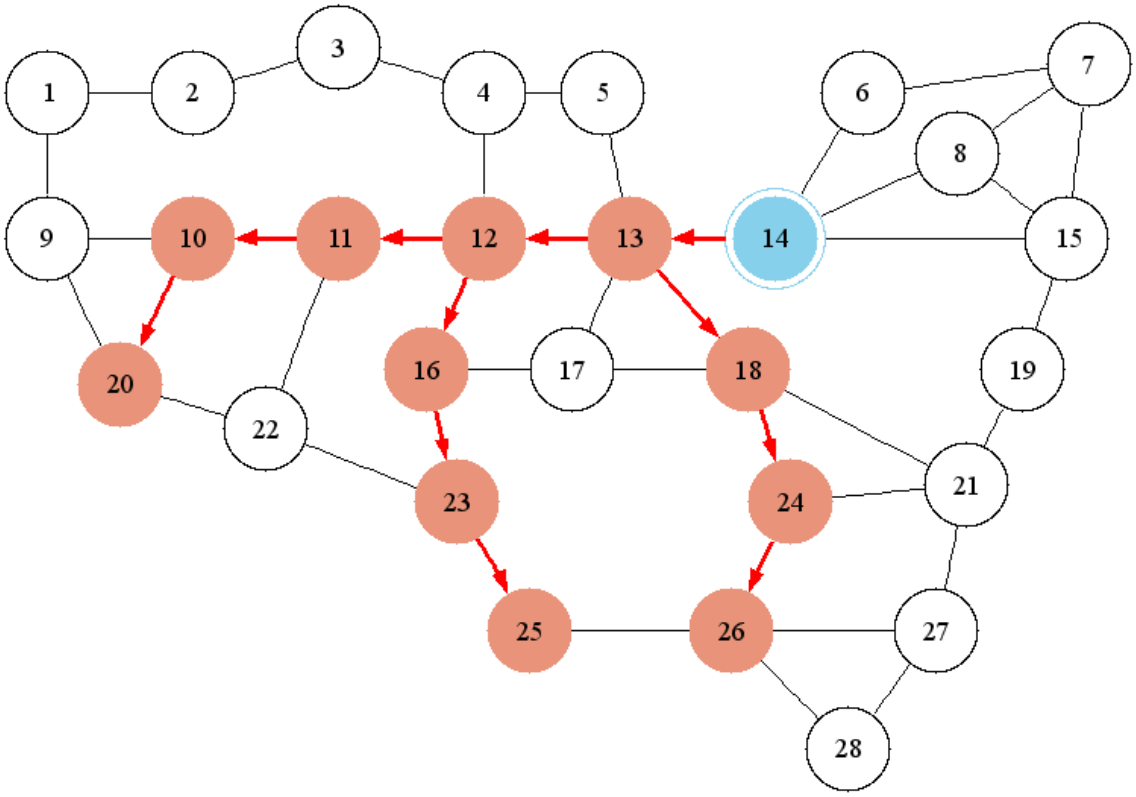

Figure 3.1 A Multicast Session

If we change the value of threshold to 3 for the same session with the same source and same set of destinations, the critical tree would be as shown in Figure 3.2. The nodes in dark salmon color indicate destinations that are part of the critical tree. The destinations that fall outside the boundary of the critical subtree are shown in green. All the edges that are part of the critical tree are shown in red. The rest of the edges are shown as green edges. It can be verified easily that if any of the components in the critical tree fails, the number of destinations that will stop receiving power from the source would be at least 3.

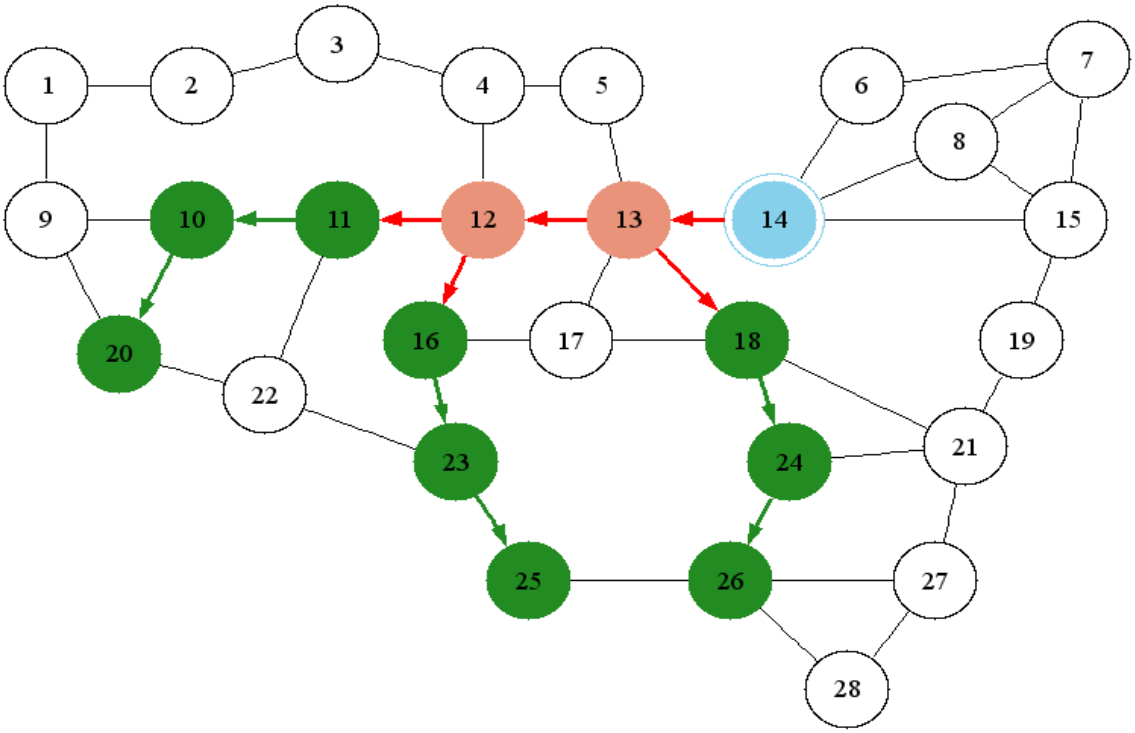

Figure 3.2 A Critical Tree

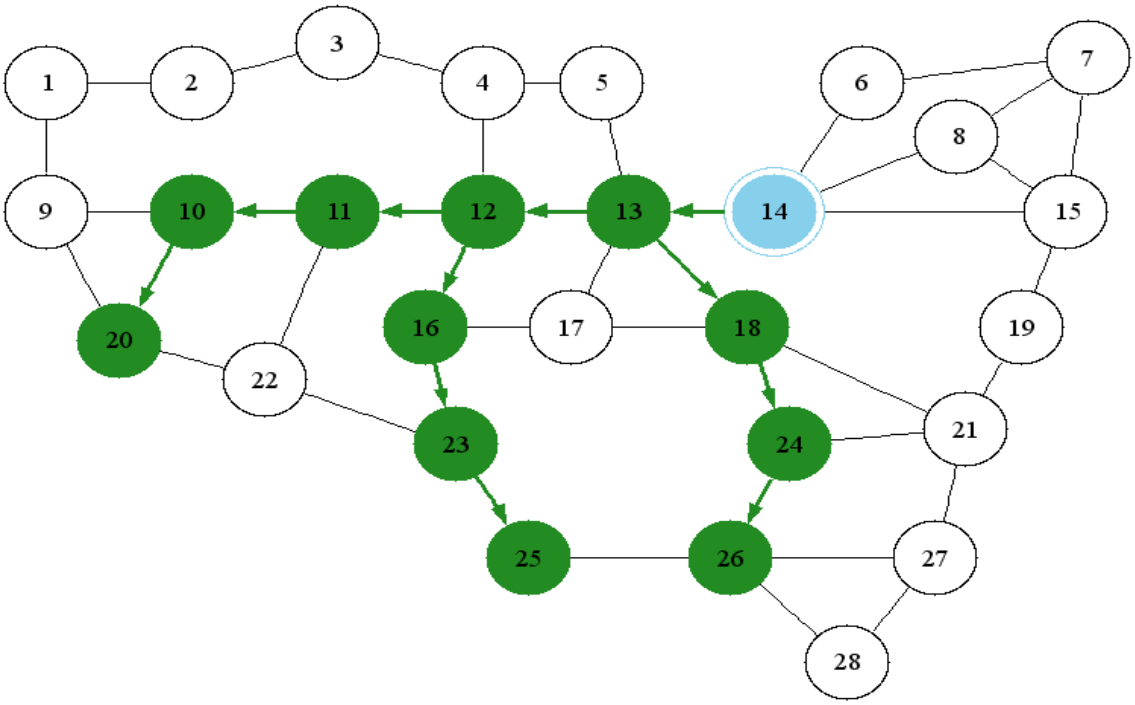

Figure 3.3 A Primary Multicast Session Without Critical Tree

We can provide a completely reactive recovery to the primary session by setting the threshold to 12. In such a case there will be no critical tree for the primary session and all failures are simply handled as and when they occur. Figure 3.3 illustrates a primary session without any critical tree.

3.1.3 Assumptions

The network is assumed to be at least 2-connected. This is to ensure that any link failure will not disconnect the network. Also, the number of wavelengths available for backup paths on any fiber is assumed greater than the number of multicast sessions. This way we can use the same fiber span for protecting different multicast sessions concurrently.

3.1.4 Fault Model

Before we explain the heuristics in details, we describe the assumed fault model. The structure of fault tolerant multicast algorithms is a natural consequence of the failure types that can occur and our ability to handle them. This fault information is captured in the fault model. Unless specified otherwise, we will use the same fault model for all the heuristics. The following fault model captures the condition of single link failure.

Failure Types:

Only single link failures are considered in the heuristic and it is assumed that at any time only one failure can occur. A failed link is equivalent to removal of the link from the network.

Failure Mode:

Dynamic: All the failures become known as and when they occur.

Failure Neighborhood:

A link failure will affect all the nodes connected to it downstream.

Failure Recovery:

Some of the failures are recovered using a protection mechanism and the rest are recovered reactively. The amount of each type of recovery being provided to the primary multicast session is dependent on the value of the *threshold* parameter.

3.2 Heuristics for Protection Based Recovery

The following sections describe three heuristics for recovering from failures using a protection scheme. Please note that, it is also possible to provide part protection and part reaction based recovery to the primary multicast session, as we shall describe later.

3.2.1 Recover by Rerouting to Neighbors

In this heuristic, we first identify the critical tree. Then, we use a concept, which we call, "*logical adjacency*" on the critical tree to recover from network failures. Two nodes lying on a linear transmission path of the critical tree are logically adjacent if they are of the following types: an MC node whose split capability is used, or a destination node, or the source node of the multicast session. The subpath between two logically adjacent nodes is called a *logical segment*. An MC node whose splitting capability is not at all used will not be considered in finding the logical segments of the critical tree. Any number of nodes and links may exist between two logically adjacent nodes. These form the members of logical segment. Notice that none of the nodes in a logical segment can be a splitting node or a destination node.

This heuristic has been named as recover by rerouting to neighbor because in this approach we recover from a link failure by switching to the back up path that has been computed for the logical segment of which the failing link is a member. Logically

adjacent nodes demarcate each of these logical segments. All the links of the logical segment will have effectively a single backup path to recover from a link failure. These "logically adjacent nodes" are considered "logical" neighbors of the failing link. By providing backup path cover to logical segments, we achieve the following benefits:

1. Reusing the same network elements for providing protection to more than one segment of the primary path, rather than to each link individually. Effectively, for a group of links between logically adjacent nodes we are using the same network resources for providing protection.
2. Reducing the backup path computation time of the primary light tree by, considering backup computations at the logical segment level. This can be termed as **Segment-based Backup Path Cover**. If the number of links in the logical segments is always more than one then the protection being provided is akin to path-based recovery. If the number of links in each segment is always just one then the protection is akin to link-based recovery.

Let's consider the USA Longhaul network to illustrate the concept of "threshold". Figure 3.4 shows the network and shows a particular multicast session. The critical tree is defined for a threshold value of 1. Following is the state of the multicast session.

Source of multicast session: 13

Multicast membership group: {11, 2, 20, 23, 25, and 26}

On the primary multicast session, all the destination nodes are indicated by dark salmon color, the split nodes are indicated by gray color and the non-destination nodes are indicated in green color. All the links belonging to one logical segment are indicated in a single color.

Let's try to find the segments on the path 13-12-16-23-25-26. *<13 – 12>* forms a segment because 13 is the source node and 12 is an MC "splitting" node. *<12 – 23>* form another segment because 23 is a destination node and none of the nodes between 12 and 23 are either splitting nodes or destination nodes. Hence, the subpath 12-16-23 is considered to be one logical segment as *<12 – 23>*. Similarly, *<23 – 25>* form a logical segment because both of them are destination nodes. The same rationale qualifies for the segment *<25 – 26>*. If we consider the path 12-11-10-2 to find the logical segments, we see that since none of the nodes between 11 and 2 are either destination or splitting nodes, therefore, *<11 – 2>* forms a logical segment. Another logical segment is formed by *<12 – 11>*.

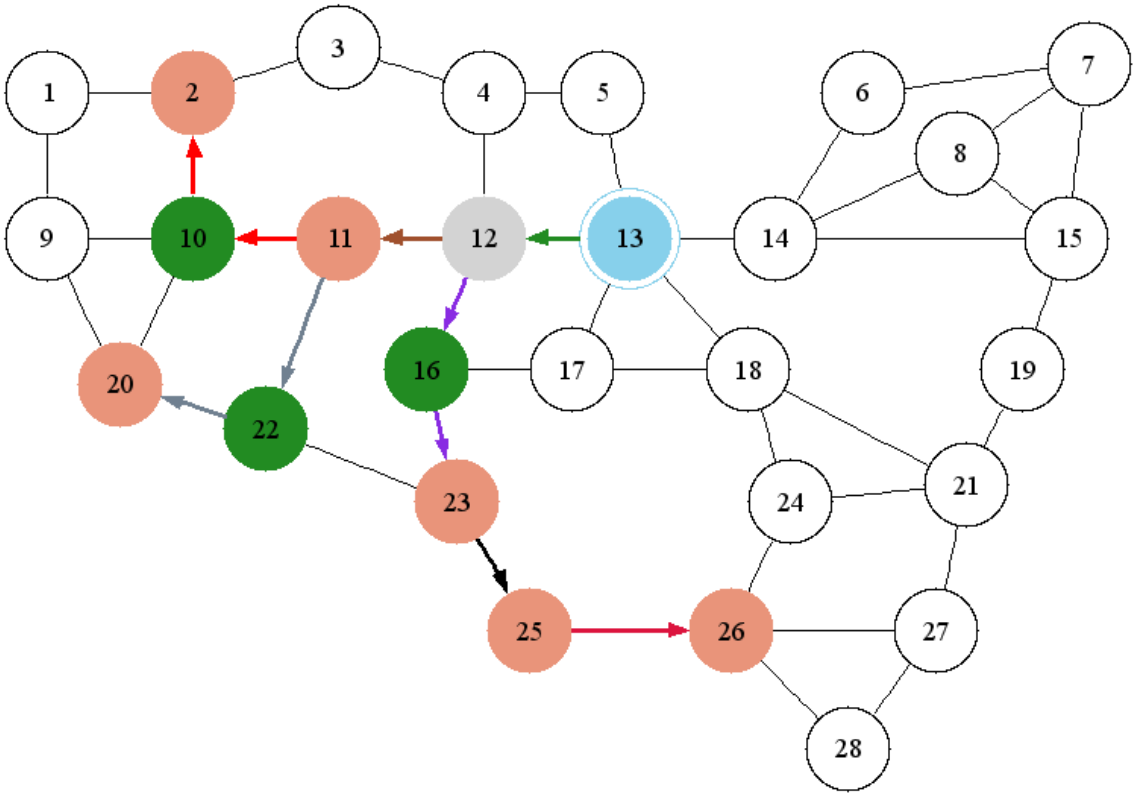

Figure 3.4 A Multicast Tree with the Identified Logical Segments

As we can see there are total of 7 logical segments in the primary session depicted in Figure 3.4, with source as node 13. These logical segments are: *< 13 – 12 >, < 12 – 11 >, < 12 – 23 >, < 23 – 25 >, < 25 – 26 >, < 11 – 20 >, < 11 – 2 >*.

3.2.1.1 Algorithm: Recover by Rerouting to Neighbors

Definitions: DEST_SET is the set of destination nodes in the multicast group. USED_SET consists of the MC nodes that have been exhaustively used or the MI nodes that have already been used as non-leaf nodes. CANDIDATE_SET is the set of nodes in current multicast tree, MCTREE, to which the new destinations can be routed. MCTREE is a single multicast session and it may not include all the members of DEST_SET. MCFOREST is the union of all MCTREE's. All the MCTREE's in MCFOREST must cover the complete set of nodes in DEST_SET. AVOID_NODES is the set of nodes that are members of a logical segment.

Input: A graph $G = (V, A)$, representing the network of MC-OXCs, a source $s \in V$, a destination set $M \subseteq V$, *powerThreshold*, a criticality threshold referred to as *criticalThreshold* [for computing the critical tree from the primary multicast tree.]

Output: A set of backup paths for the logical segments, which protect the primary session formed using s and M.

1. **begin**
2. **while** (DEST_SET is not empty)
3. Initialize MCTREE to be an empty tree.
4. Find shortest path of every node in CANDIDATE_SET to every node in DEST_SET, such that no node on this path is in the USED_SET.
5. **if** (more than one such path exist) **then**

6. Select a path P(u, v) among the shortest paths. /* See the work [1] for further details about this step. */

7. Add P(u, v) to MCTREE.

8. Update the sets USED_SET, DEST_SET and CANDIDATE_SET based on the nodes in P(u,v) that has been added, into the tree.

9. **end-if**

/*At the onset of this step, all the destinations have been added to the multicast tree and is ready to undergo backtracking technique. */

10. Calculate the minimal power being delivered to any destination in the multicast tree, P_{min} constructed so far.

11. Apply backtracking /*See the work in [1] for further details about this step*/ to see if the minimal power being delivered in the multicast tree can be improved beyond P_{min}. If the power can be improved then we retain the changes due to backtracking, otherwise we discard the changes.

12. Find all the nodes with power < *powerThreshold*. Add these destinations to the set DEST_SET and update the sets, CANDIDATE_SET and USED_SET.

13. Add MCTREE to MCFOREST.

14. **end-while**

/* Now at this point we have a set of trees as multicast sessions in the forest that we call, MCFOREST. */

15. **for** each MC Tree, MCTREE, in the forest, MCFOREST

/* Calculate the Critical Tree. */

16. **for** each node n in MCTREE

17. Compute a breadth first search from node n.

18. Compute the number of destination nodes found in the breadth first search from node n. Let this number be denoted as affectedDestinations.

19. **if** (affectedDestinations > *criticalThreshold*) **then**

20. Node n is a critical node. Update the adjacency list being maintained for the critical tree.

21. **end-if**

22. **end-for**

/* Computing the logical segments of the critical tree */

23. **for** each path to the leaf nodes in the critical tree

24. Divide the path into logical segments.

25. **end-for**

26. For each logical segment calculate the backup path. While computing the backup path keep a list of nodes that form the part of the segments found as AVOID_NODES. While computing the shortest path for the end points of the segment try to find a path that doesn't contain any nodes in the set AVOID_NODES or if necessary should contain the minimal number of such nodes.

27. **end-for**

28. **end-begin**

Note: One way to improve upon the *for* loop in step 15 is to find the affected number of destinations upon failure of a node by starting from leaf nodes and moving towards the source of the transmission. The first encounter of a critical node on a path from a

particular leaf node to the source indicates that all the parents from there on are part of the critical tree. Thus, we can terminate the process for that path and consider the next path from some other leaf node to the source. This process can be repeated for all the leaf nodes to construct the critical tree. This way we can narrow down the number of nodes being checked for criticality.

3.2.1.2 Handling Single Link Failure

Each link will belong to a particular logical segment of the critical tree. The first step in handling a single link failure is to establish the membership of the link to a particular logical segment that has been protected by a corresponding backup path. Since this membership is already known at the time of backup path computations, hence the recovery is simply done by switching the transmission to the backup path. Please note that on some of the links the direction of transmission may be changed in the process.

3.2.1.3 An Example

The example illustrated here uses the 28-node USA Longhaul network. We will use the multicast session in Figure 3.4 to illustrate the protection by this scheme. Figure 3.5 shows the state of the network under the condition where the link 11-10 has failed. For convenience we summarize the state as follows:

Source of multicast session: 13

Destination Set: {11, 2, 20, 23, 25, and 26}

Failed link: 11-10

The link 11-10 falls in the segmented path cover of logical segment 11-2.

The backup path for this segment is 11-12-4-3-2. Notice that link 11-12 has been included in the backup path because we utilize the property of directed-link disjointness.

In Figure 3.5 the back up path that will be used to recover from the failure is 12-4-3-2 and the failed link 11-10 has been removed from the network. As before all the destination nodes are indicated by dark salmon color, the split nodes are indicated by gray color and the non-destination nodes are indicated in green color.

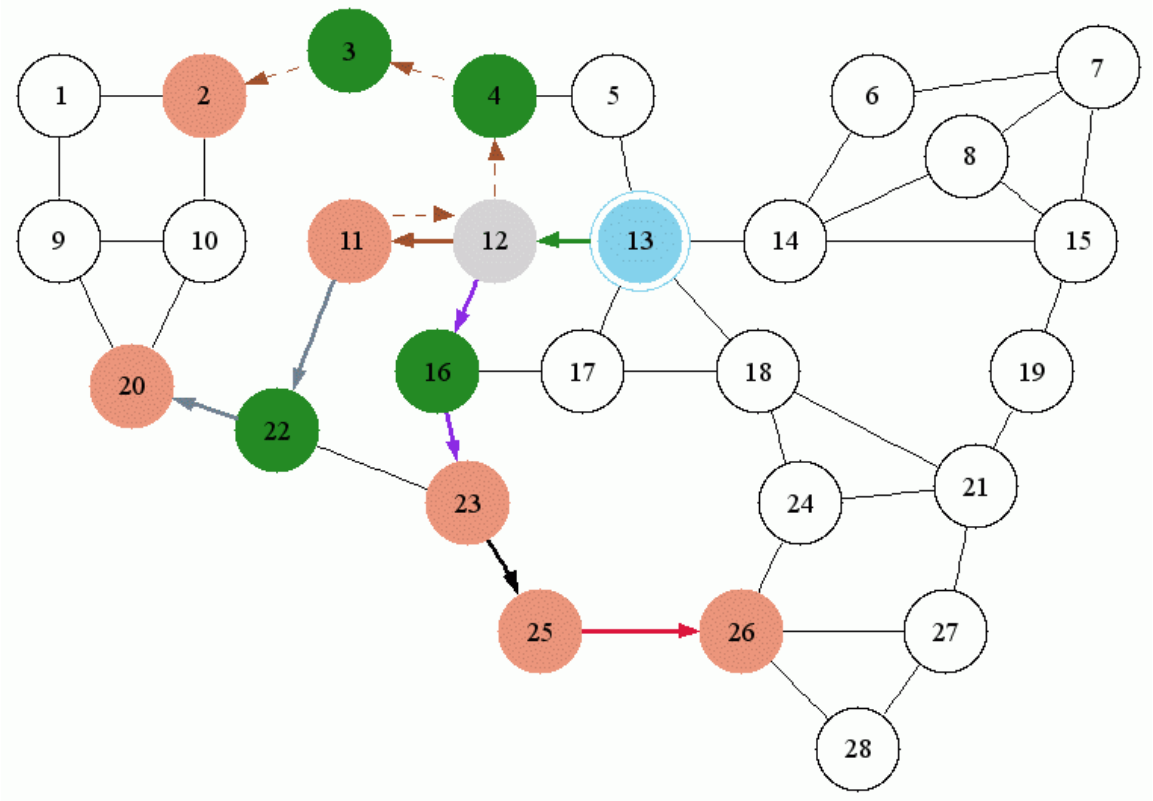

Figure 3.5 Illustration of Recover by Rerouting to Neighbors

3.2.2 Recover by Rerouting To Source

In this heuristic, we first identify the critical tree. A backup path is constructed from the source node to each of the destinations and to each multicast capable and splitting node in the critical tree. In constructing the backup paths we avoid the links on the critical tree whose failure can affect transmission to these nodes.

3.2.2.1 Algorithm: Recover by Rerouting to Source

Definitions: DEST_SET is the set of destination nodes in the multicast group. USED_SET consists of the MC nodes that have been exhaustively used or the MI nodes that have already been used as non-leaf nodes. CANDIDATE_SET is the set of nodes in current multicast tree, MCTREE, to which the new destinations can be routed. MCTREE is a single multicast session and it may not include all the members of DEST_SET. MCFOREST is the union of all MCTREE's. All the MCTREE's in MCFOREST must cover the complete set of nodes in DEST_SET.

Input: A graph $G = (V, A)$, representing the network of MC-OXCs, a source $s \in V$, a destination set $M \subseteq V$, *powerThreshold*, a criticality threshold referred to as *criticalThreshold* [for computing the critical tree from the primary multicast tree.]

Output: A set of backup paths for the logical segments, which protect the primary session formed using s and M.

1. **begin**
2. **while** (DEST_SET is not empty)
3. Initialize MCTREE to be an empty tree.
4. Find shortest path of every node in CANDIDATE_SET to every node in DEST_SET, such that no node on this path is in the USED_SET.
5. **if** (more than one such path exist) **then**
6. Select a path P(u,v) among the shortest paths. /*See the work [1] for further details about this step. */
7. Add P(u, v) to MCTREE.

8. Update the sets USED_SET, DEST_SET and CANDIDATE_SET based on the nodes in P(u,v) that has been added, into the tree.

9. **end-if**

/*At the onset of this step, all the destinations have been added to the multicast tree that is ready to undergo backtracking. */

10. Calculate the minimal power being delivered to any destination in the multicast tree, P_{min} constructed so far.

11. Apply backtracking /*See the work in [1] for further details about this step*/, to see if the minimal power being delivered in the multicast tree can be improved beyond P_{min}. If the power can be improved then we retain the changes due to backtracking, otherwise we discard the changes.

12. Find all the nodes with power < *powerThreshold*. Add these destinations to the set DEST_SET and update the sets, CANDIDATE_SET and USED_SET.

13. Add MCTREE to MCFOREST.

14. **end-while**

/* Now at this point we have a set of trees as multicast sessions in the forest store that we call, MCFOREST. */

15. **for** each MC Tree MCTREE in the forest, MCFOREST

/* Calculate the Critical Tree. */

16. **for** each node n in MCTREE

17. Compute a breadth first search from node n.

18. Compute the number of destination nodes found in breadth first search from node n. Let this number be affectedDestinations

19. **if** (affectedDestinations > *criticalThreshold*) **then**

20. Node n is a critical node. Update the adjacency list being maintained for the critical tree.

21. **end-if**

22. **end-for**

23. Compute the paths from the source *s*, to all the destination, and splitting in the critical tree.

24. **end-for**

25. **end-begin**

3.2.2.2 Handling Single Link Failure

Upon the failure of a single link find the first node downstream from it in the primary tree, which is a destination or a splitting node. Resume the transmission using the path to that node from the source *s*.

3.2.2.3 An Example

Consider the USA Longhaul network and the following situation:

Source of multicast session: 13

Destination set: {11, 2, 20, 23, 25, and 26}

Failed link: 11-10

This example is illustrated in Figure 3.6

Upon failure of link 11-10, node 2 will be prevented from receiving the transmission. In this case we switch to the backup path 13-5-4-3-2 to continue receiving the transmission at node 2. In Figure 3.6 the back up path that will be used to recover from the link failure

has been indicated in dotted lines while the failed link 11-10 has been removed from the network.

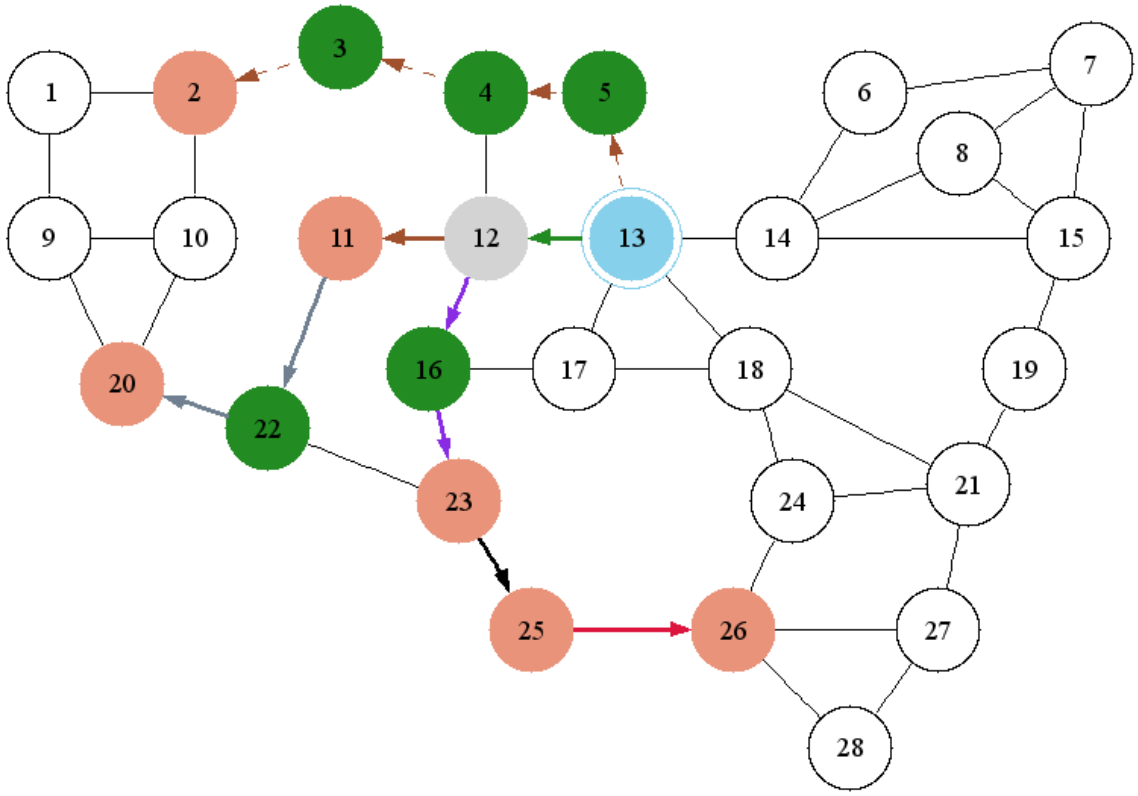

Figure 3.6 Illustration of Recover by Rerouting to Source

3.2.3 Recover by Re-Computation

Here we introduce a new approach for recovery. This approach is a more elaborate method for finding the backup paths as compared to the two previous methods. In this approach we consider all the possible single link failures one by one and come up with alternate routing options. The idea is to:

1. Create a primary light tree.
2. Fail each of the links on the primary tree one be one and
    a. For each of the potential link failures gather the information on the destinations that would be affected and,

b. Create a new multicast tree for the affected set of destination nodes that does not use the failed link.

So we will have two trees when the recovery is done:

1. The intact portion of the primary tree after the single link failure and;
2. An alternate transmission tree for the affected destinations.

3.2.3.1 Algorithm: Recover by Re-Computation

Definitions: DEST_SET is the set of destination nodes in the multicast group. USED_SET consists of the MC nodes that have been exhaustively used or the MI nodes that have already been used as non-leaf nodes. CANDIDATE_SET is the set of nodes in current multicast tree, MCTREE, to which the new destinations can be routed. MCTREE is a single multicast session and it may not include all the members of DEST_SET. MCFOREST is the union of all MCTREE's. All the MCTREE's in MCFOREST must cover the complete set of nodes in DEST_SET.

Input: A graph $G = (V, A)$, representing the network of MC-OXCs, a source $s \in V$, a destination set $M \subseteq V$, *powerThreshold*,

Output: A set of backup paths for the logical segments, which protect the primary session formed using s and M.

1. **begin**

2. **while** (DEST_SET is not empty)

3. Initialize MCTREE to be an empty tree.

4. Find shortest path of every node in CANDIDATE_SET to every node in DEST_SET, such that no node on this path is in the USED_SET.

5. **if** (more than one such path exist) **then**

6. Select a path P(u,v) among the shortest paths. /*See the work [1] for further details about this step.*/

7. Add P(u, v) to MCTREE.

8. Update the sets USED_SET, DEST_SET and CANDIDATE_SET based on the nodes in P(u,v) that has been added, into the tree.

9. **end-if**

/*At the onset of this step, all the destinations have been added to the multicast tree that is ready to undergo backtracking. */

10. Calculate the minimal power being delivered to any destination in the multicast tree, P_{min} constructed so far.

11. Apply backtracking /*See the work in [1] for further details about this step*/, to see if the minimal power being delivered in the multicast tree can be improved beyond P_{min}. If the power can be improved then we retain the changes due to backtracking, otherwise we discard the changes.

12. Find all the nodes with power < *powerThreshold*. Add these destinations to the set DEST_SET and update the sets, CANDIDATE_SET and USED_SET.

13. Add MCTREE to MCFOREST.

14. **end-while**

/* Now at this point we have a set of trees as multicast sessions in the forest store that we call, MCFOREST. */

15. **for** each MC Tree, MCTREE in the forest, MCFOREST

16. **for** each link *li* in MCTREE

17. Find all the affected destinations that would be affected if that link (*li*) were to fail.

18. Construct a new multicast tree (disjoint of the failed link *li*) for the affected components from the source.

19. Store the alternate MC tree in a lookup table against the link *li*.

20. end-for

21. end-for

22. end-begin

3.2.3.2 Handling Single Link Failure

Upon discovering a failed link use a look up to retrieve the computed alternate MC tree. Start using the retrieved alternate MC tree for recovering from that link failure.

3.2.3.3 An Example

As before, consider the USA Longhaul network and the following situation:

Source of multicast session: 13

Destination set: {11, 2, 20, 23, 25, and 26}

Failed link: 12-11

Upon failure of link 12-11, nodes 2, 11 and 20 will be deprived from receiving the transmission. The leftover of the primary MC session is depicted in Figure 3.7 and the backup MC tree that will be used to recover are depicted in Figure 3.8. For clarity, the affected destinations are also depicted in Figure 3.7.

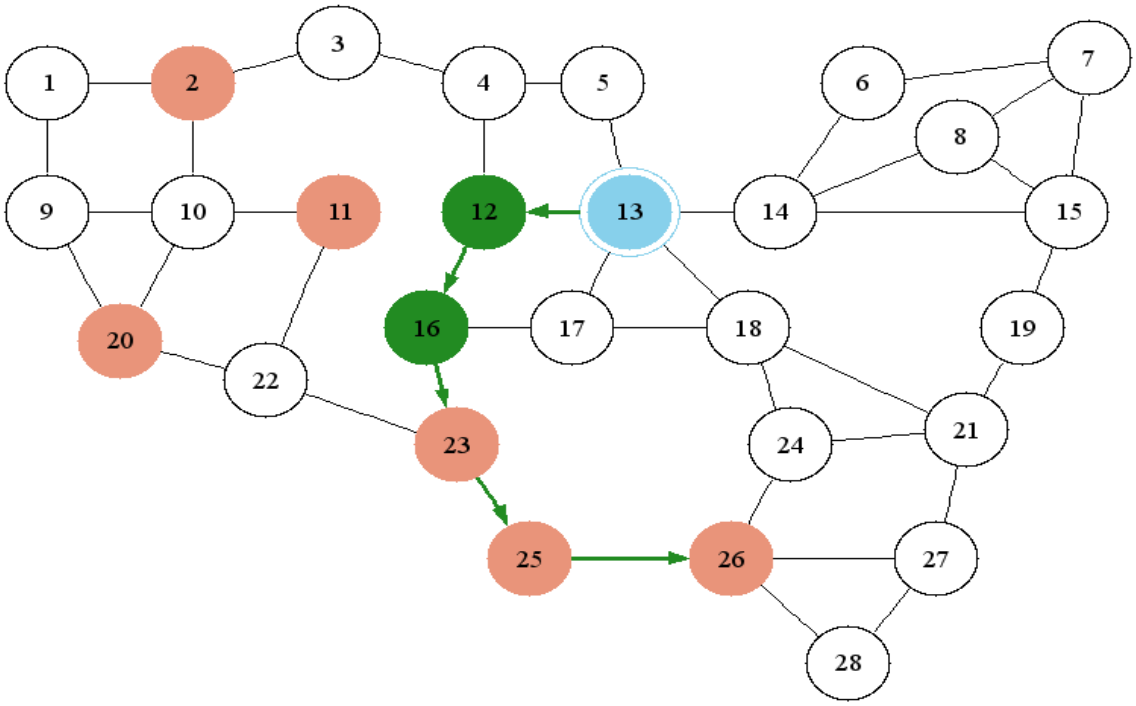

Figure 3.7 Affected Destinations Due to Link Failure

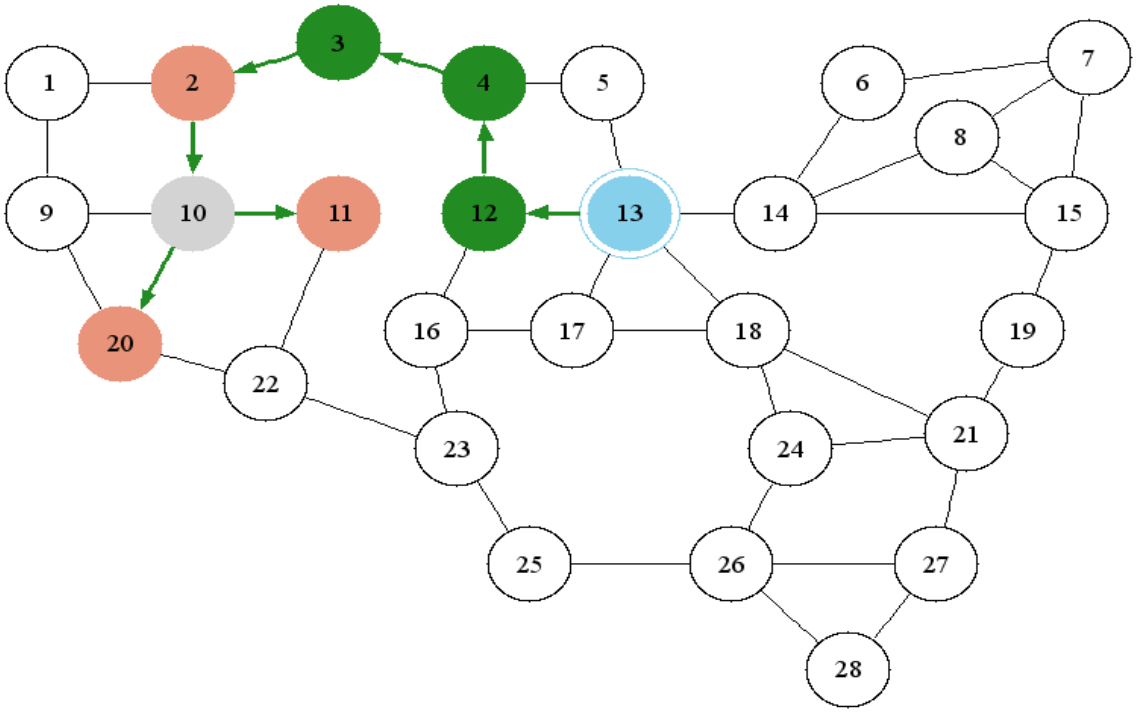

Figure 3.8 Alternate MC Tree

3.3 Heuristics for Reaction Based Recovery

The following section describes two heuristic algorithms for recovering from failures using reactive schemes. Please note that, it is possible to use a hybrid approach that provides part protection-based and part reaction-based recovery. The basis of these algorithms is that whenever a link failure occurs, it disturbs the primary transmission and breaks it into two trees. We approach the problem of connecting these two trees using two heuristics as described next.

3.3.1 Recover by Rerouting to Any

Upon failure of a link the primary multicast transmission is split into two parts: the intact portion of the main multicast tree and a disconnected subtree. The idea behind this heuristic is to connect these two disconnected trees to resume transmission of signals to the affected destinations. In this method, the first available node in the disconnected subtree, which has lost transmission, is connected to the closest useful node on the main multicast tree. Typically, the non-critical portion of the primary tree associated with the approaches "*Recover by rerouting to neighbors*" and "*Recover by rerouting to source*" (described in Section 3.2) will be recovered using this scheme.

3.3.1.1 Algorithm: Recover by Rerouting to Any

Definitions: CANDIDATE_SET is the set of nodes in MCTREE to which the new destinations can be routed.

Input: A graph $G = (V, A)$, representing the network of MC-OXCs, a source $s \in V$, a destination set $M \subseteq V$.

Output: A backup path that is computed dynamically as the failure is discovered.

1. **begin**
2. ***When a link failure is detected* {**
3. Find all the destinations and MC capable nodes that are affected by the failed link.
4. Sort the nodes found in step 3 according to their distance from the failure, in a list *OrderedList*
5. Attach the first possible node in *OrderedList* by a shortest distance to any of the useful nodes in CANDITATE_SET.
6. }
7. **end-begin**

3.3.1.2 Handling Single Link Failure

To resume transmission to the disconnected branch we start by considering connecting the root of the disconnected subtree. If we cannot find a path from the root of the disconnected subtree then the next node in the disconnected subtree that is a splitting or a destination node is considered. For such a node we try finding a shortest path to a node in the set CANDIDATE_SET. Non-MC and non-destination nodes of the disconnected subtree are not considered for this purpose. The path from a node of the disconnected subtree to all the nodes in CANDIDATE_SET is considered and the shortest one among them is chosen. Recovery is then achieved by switching to this path. It should be noted that we may have to reverse the direction of transmission on some links so as to cover all the destinations in the disconnected subtree and achieve complete recovery.

3.3.1.3 An Example

The example uses the USA Longhaul network and assumes the multicast session of Figure 3.4. The following is the state of the multicast session depicted in Figure 3.4.

Source of multicast session: 13

Destination set: {11, 2, 20, 23, 25, and 26}

Failed link: 12-16

Upon link failure, we try to connect the first available node in the subtree affected by the failure to the intact part of the primary tree. We will consider the nodes 16, 23, 25, and 26 in that order (See Step 4 of the algorithm). We try finding the shortest paths to nodes in the CANDIDATE_SET which in this case contains the nodes 13, 22, 20, 10, and 2 of the

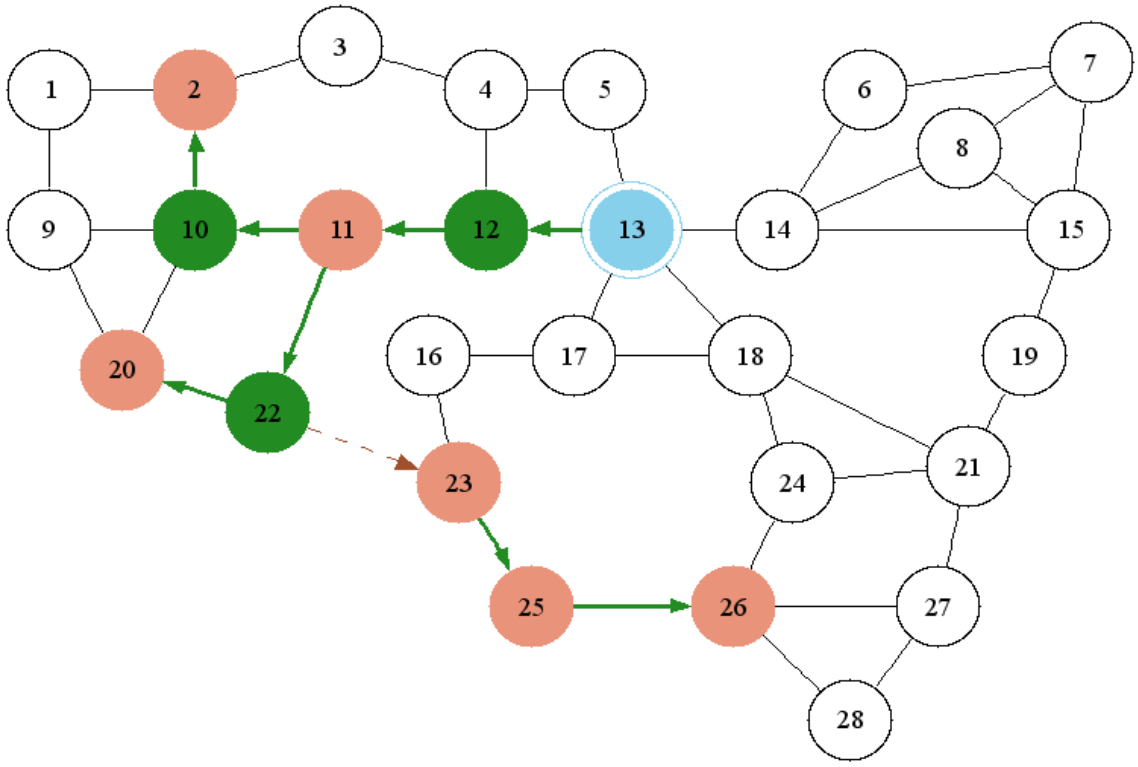

Figure 3.9 Illustration of Recover by Rerouting to Any

intact part of the primary tree. The backup path that will be used to recover from the failure has been indicated in dotted lines and the failed link 12-16 has been removed from the network in Figure 3.9. In this case since node 16 is a non-destination node, it does not receive the signal. Instead node 23 connects to node 11 via node 22 for receiving the transmission.

3.3.2 Recover by Rerouting to Leaf

Upon failure of a link, the primary multicast transmission is split into two parts: the intact portion of the main multicast tree and a disconnected subtree. The basic idea of this heuristic is also to connect these two disconnected subtrees so as to resume transmission of signals to the affected destinations. In this approach, all the leaf nodes that are affected by the link failure are attached to the intact part of the primary MC Tree. It should be noted that we may have to reverse the direction of transmission on some links so as to cover all the destinations in the disconnected tree and achieve complete recovery. Typically, the non-critical portion of the primary tree in the approaches "*Recover by rerouting to neighbors*" and "*Recover by rerouting to source*" will be recovered using this scheme.

3.3.2.1 Algorithm: Recover by Rerouting to Leaf

Definitions: CANDIDATE_SET is the set of nodes in MCTREE to which the new destinations can be routed.

Input: A graph $G = (V, A)$, representing the network of MC-OXCs, a source $s \in V$, a destination set $M \subseteq V$.

Output: A backup path that is computed dynamically as the failure is discovered.

1. **begin**
2. ***When a failure is detected*** **{**
3. Find all the leaf nodes in the disconnected subtree(all the leaf nodes are destination nodes).
4. Attach each such leaf node to the nearest node in the CANDIDATE_SET via a shortest path.
5. **}**
6. **end-begin**

Note: The direction of transmission after the recovery may have to change on some links to continue transmission to the affected nodes. For a particular disconnected subtree the number of leaf nodes may be more than one. In such a case, as we proceed connecting each such leaf node to a node in the CANDIDATE_SET we also update the nodes in the set CANDIDATE_SET.

3.3.2.2 Handling Single Link Failure

In this approach, to recover from a single link failure, each of the leaf nodes in the disconnected subtree is connected to the nearest node in CANDIDATE_SET of the intact part of the primary transmission tree via a shortest path. In the process we may have to reverse the direction of transmission on some of the links to continue transmission to the affected nodes. This is illustrated in the Figure 3.10.

3.3.2.3 An Example

The example uses the USA Longhaul network and the multicast session of Figure 3.4 to illustrate recovery using this scheme. The following is the state of the multicast session depicted in Figure 3.4.

Source of multicast session: 13

Destination set: {11, 2, 20, 23, 25, and 26}

Failed link: 12-16

Upon link failure, we try to connect all the affected leaf nodes in the part of the tree affected by the failure to the intact part of the primary tree. The direction of the transmission is changed appropriately to receive the signal. In this case, the only leaf node in the path affected by the failure of the link 12-16 is node 26.

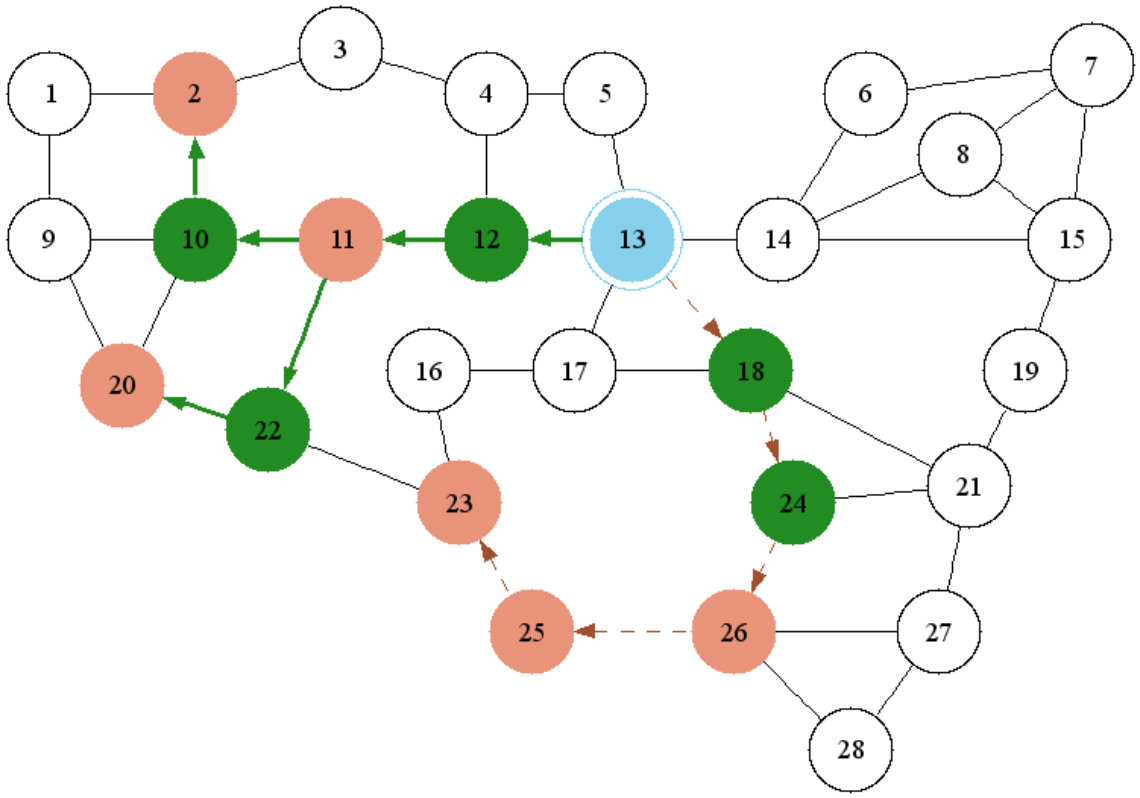

Figure 3.10 Illustration of Recover by Rerouting to Leaf

Based on the shortest path heuristic, node 26 is connected to node 13 of the CANDIDATE_SET. To continue transmitting the signal to nodes 25 and 23, the direction of transmission is changed. The backup path that will be used to recover from the failure has been indicated in dotted lines and the failed link 12-16 has been removed from the network in Figure 3.10.

Chapter 4

Results and Discussions

This chapter evaluates the behaviour of each of the algorithms in Chapter 3 both in standalone mode and in combination as illustrated in Figure 4.1. For that purpose we have designed and implemented a simulator of some 3000 lines. In Chapter 5 detailed information regarding the simulator and its setup will be given.

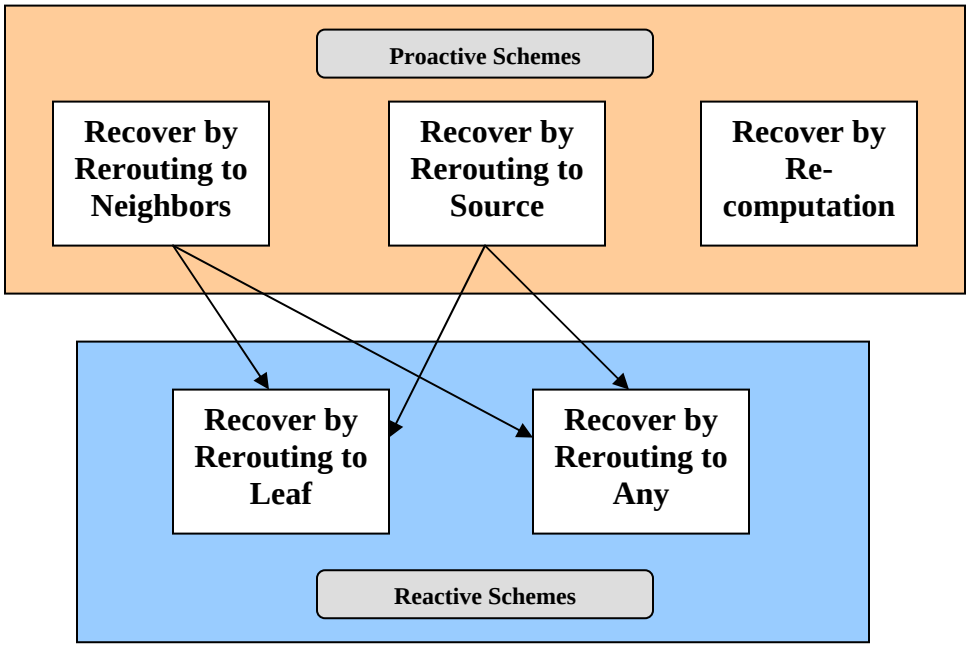

Figure 4.1 Recovery Heuristics and their Combinations

Two of the proactive measures will be combined with two reactive measures to arrive upon a total of four combinations. The dark arrows in Figure 4.1 indicate the various combinations that will be considered in the simulations. The proactive scheme *Recover by recomputation*, was not considered for combinations because of its high time complexity.

4.1 Performance Metrics

The following parameters are utilized in comparing the various schemes, either by themselves, or in combinations using the concept of threshold.

a. **Computation time**: This parameter tells us the time taken to execute the heuristics and compute the backup paths.

b. **Average number of links**: This metric measures usage of network resources by the studied heuristic. It indicates the average number of links that will be used in the backup paths for recovery purposes.

c. **Average power delivered**: This measure gives a fair idea of how effective an algorithm is in terms of the power delivered to destinations after recovering from the link failure.

d. **Percentage change in the average power**: This measure gives an idea of the percentage change in the power received in the recovered multicast tree as compared to the power delivered in the original multicast session

The above parameters are measured versus the following variables for different heuristics (combinations):

a. **Multicast membership size**: The multicast membership size is varied from 20 to 40 percent of the total number of nodes in the network.

b. **Threshold**: We will vary the threshold from 1 to a value equal to the multicast membership size.

The parameters for complete reactive and complete proactive measures will be measured against MC membership size only. The combinations of reactive and protective schemes will be tested similarly by varying the MC membership size. While combining

the protective and reactive schemes, a threshold value is chosen such that an equal number of links is divided between protective and reactive schemes for their recovery. By setting the value of threshold we can control the number of components in the multicast session to be considered under protection and/or reaction based recovery.

For each multicast session, the metrics are evaluated by:

a. Setting the threshold for the session
b. Selecting an approach to recover from the failure
c. Considering 4 to 5 individual single link failures for each MC session and then averaging out the result for these failures. This way we increase the confidence in the metrics obtained. We vary the MC group size from 20% to 40% of the total number nodes in the network. Thus, for the 28-node USA Longhaul network we consider 6 different MC sessions with MC group sizes varying from 6 to 11 nodes. In our case 4 to 5 links in most of the MC sessions form roughly 50% of the links in the primary tree. Considering all the link failures one by one on the primary tree would have been too exhaustive and time consuming. For each MC group, we compute all the metrics for 5 different recovery algorithms. For each MC session we considered 4-5 link failures which results in computing the metrics on atleast 20 alternate MC trees (for recovering from each link failure) for all 5 algorithms. Over 6 MC different sessions this number is roughly around 120 trees which is a significant number for each metric to be computed.

Unless otherwise stated we use the USA Longhaul network or a variant thereof as the test-bed for running our simulations. The USA Longhaul is a 2-connected network with 28 nodes and 42 links.

4.2 Computation Time Metrics

The computation time for a recovery scheme refers to the execution time of the heuristic. This gives a measure of the time complexity of these heuristics. The proactive heuristics have been studied as one group and the reactive schemes have been studied as another group. For a multicast session with a particular MC group size, this metric indicates the average computation time used for computing backup paths for 4 to 5 different single link failures in that multicast session.

4.2.1 Results for Proactive Schemes

The computation time for both the reactive and proactive schemes is studied by varying the multicast membership size. The behavior obtained for the proactive schemes is plotted in Figure 4.2. As we can clearly see the time to recover using the scheme *Recover by recomputation* is more than that of any other scheme. Further the time grows quickly as the MC membership size grows. This is simply because for the heuristic (ALGORITHM 2 in [1]) to generate power efficient multicast sessions for the affected destinations (upon a link failure) large amount of time are needed to compute alternate MC trees for each possible single link failure. Also, as the number of destinations increases, the number of links in the MC tree also increases which consequently gives rise to larger computation times. As seen in Figure 4.2, the *Recover by rerouting to source* scheme takes a slightly more time to execute than the *Recover by rerouting to neighbors* scheme. This behavior may be attributed to the longer paths to be found from

the destination nodes to the source node in the former compared to shorter paths found between logically adjacent nodes in the *Recover by rerouting to neighbors* scheme.

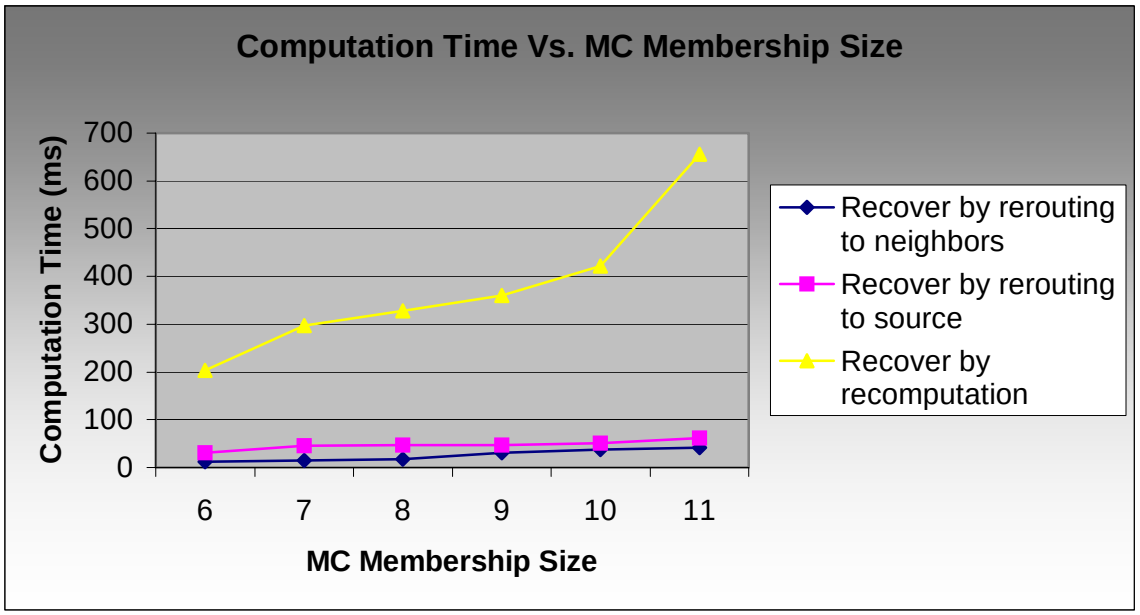

Figure 4.2 Computation Time vs. MC Membership Size for Proactive Schemes

4.2.2 Results for Reactive Schemes

The computation times for the reactive schemes are depicted in Figure 4.3. The schemes do not show marked difference in their computation times, but between the two schemes, *Recover by rerouting to any* runs faster than *Recover by rerouting to leaf*. This is because in the former scheme the disconnected subtree is connected to the closest possible useful node in the intact part of the primary MC tree. Also, as can be seen from the plot the computation time generally increases with the increase in MC membership size.

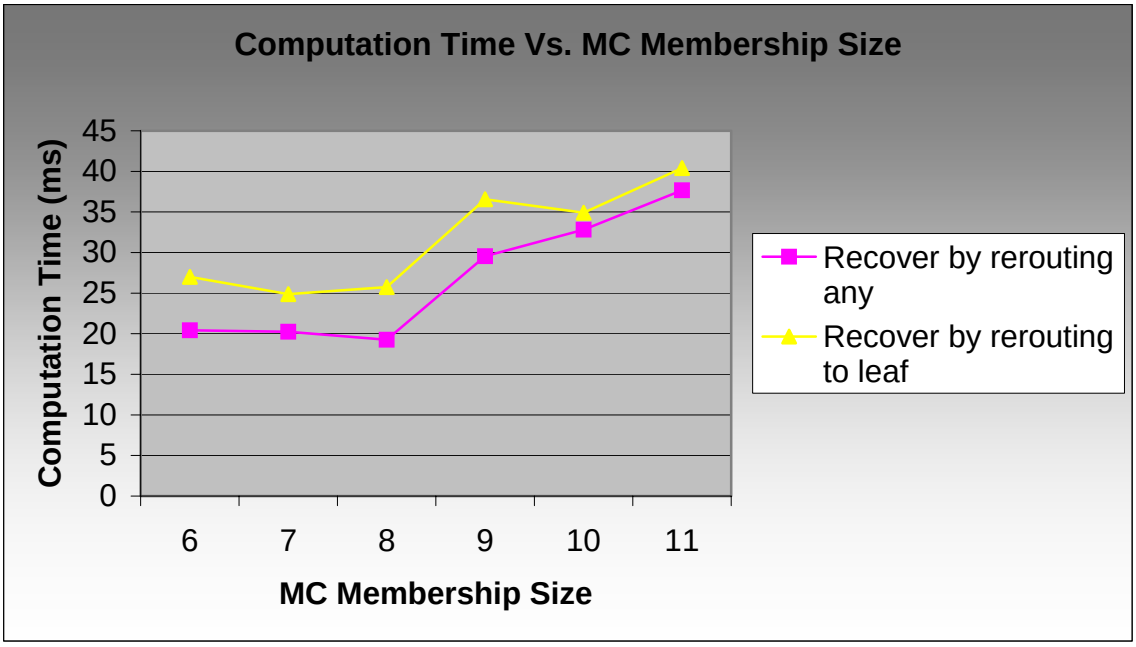

Figure 4.3 Computation Time vs. MC Membership Size for Reactive Schemes

4.2.3 Results for Hybrid Schemes

A multicast session can be recovered by proactive schemes, reactive schemes or both schemes in a combination. To accomplish a hybrid recovery model, we use the value, *threshold*, to divide the primary multicast session into critical and non critical portions, which are subsequently handled by proactive and reactive recovery schemes, respectively. The plot in Figure 4.4 depicts the behavior of the proactive and reactive schemes with respect to computation time utilized in various combinations, as discussed earlier.

The symbols P1, P2, R1, R2 in the plot and elsewhere in this chapter, are used to denote the protective and reactive schemes. **P1** refers to r*ecover by rerouting to neighbors*, **P2** refers to *recover by rerouting to source*, **R1** refers to *recover by rerouting to any*, and **R2** refers to *recover by rerouting to leaf*. As it can be seen from Figure 4.4, the behavior towards the left predominantly reflects the effect of proactive measures

much more than the reactive measures. As the threshold value increases the effect of reactive measures become more dominant. We can actually comprehend the behavior of the plot in Figure 4.4, based on the plots obtained in Sections 4.2.1 and 4.2.2. The MC membership size is fixed at 11 in these simulations.

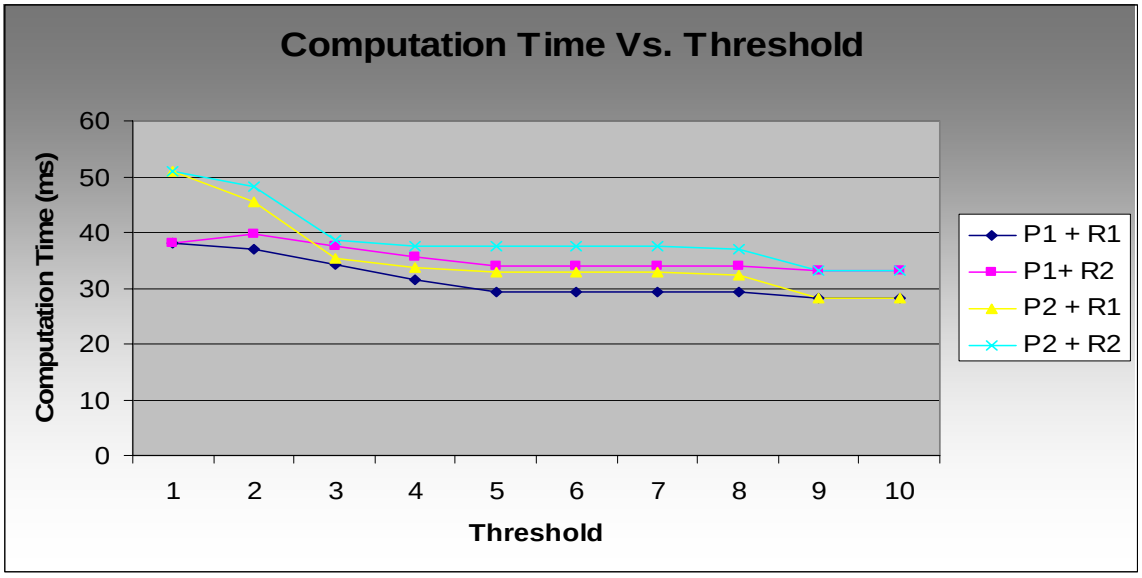

Figure 4.4 Computation Time vs. MC Membership Size for Hybrid Schemes

From the computation time plots of all the schemes we know that among the reactive measures, R2 requires more computation time than R1 and among the proactive measures, P2 requires more computation time than P1. We also note that the reactive schemes in general are faster than the proactive schemes.

Initially, when the threshold is set to 1 (i.e. no reactive cover is being provided), R1 or R2 are irrelevant and schemes using P2 take more time than schemes with P1. As the threshold value is increased, the time of execution should go down as the number of components being recovered reactively increases. Thus, as the threshold increases, the affect of R1 and R2 on computation time becomes more significant and the hybrid schemes with R2 take more time than the hybrid schemes with R1. The behavior of the

hybrid scheme is fully rationalized based on the behavior noted for each of the recovery schemes individually.

4.3 Average Number of Links Metrics

The amount of network resources that will be used to recover from a failure is an important factor to consider if the network resources are to be used judiciously and effectively. For a multicast session with a particular MC group size, this metric indicates the average number of links used for 4 to 5 different single link failures in that multicast session.

4.3.1 Results for Proactive Schemes

The average number of links that will be used by the *recover by recomputation* scheme is more than that for any of the other methods because with this scheme the alternate MC trees for the failures tend to utilize more links in general. The average number of links that will be used by *recover by rerouting to source* is more than that of *recover by rerouting to neighbors*. This behavior is depicted in Figure 4.5. This trend can be explained as follows. For each of the nodes in the former method we find a path from the source, whereas in the latter case we find backup path between logically adjacent nodes. Even though we find backup path explicitly to either the source or for the logical segment, we may still attach the affected node to the first feasible node which is on this backup path and also on the primary multicast tree. However, the length of the path found by the *rerouting to source* scheme tends to be longer than the ones found by the *rerouting to neighbors* scheme. The plot shows that the *recover by rerouting to source* scheme uses an average number of links that is higher than that for the *recover by rerouting to neighbors* scheme.

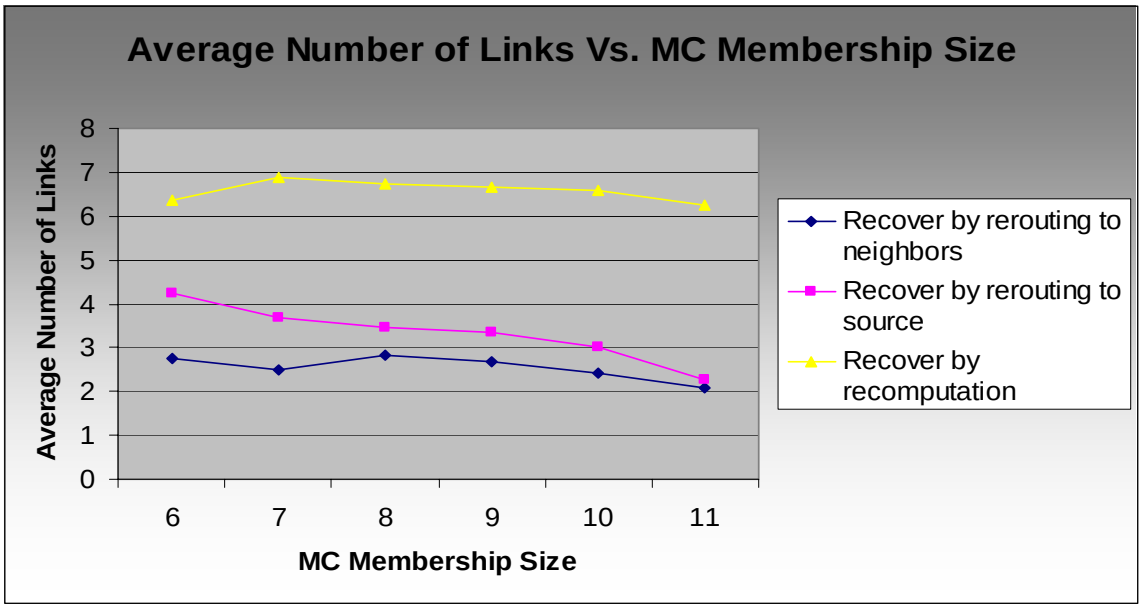

Figure 4.5 Average No. Of Links vs. MC Membership Size for Proactive Schemes

As the MC membership size increases, the average number of links goes down. This may be explained as follows. As the MC size increases the number of useful nodes on the primary session that can be connected in reactive recovery methods also increases. The path length therefore decreases as the MC membership size increases thus bringing down the average length of the recovery path.

4.3.2 Results for Reactive Schemes

Figure 4.6 shows the average number of links used in recovery versus membership size for reactive schemes. As can be seen, the number of links used by the *recover by rerouting to leaf* scheme is consistently higher than that used by the approach *recover by rerouting to any*. This is because the connection in the former case is to the leaf nodes which may be far off compared to the first available node as with the latter approach. Figure 4.6 also shows that as MC membership size increases, the average number of links goes down. This may be because as the MC size increases the number of

useful nodes on the primary session that can be connected to, in either of the two methods, also increases. The path length therefore is likely to decrease as the MC membership size increases thus bringing down the average length of recovery paths.

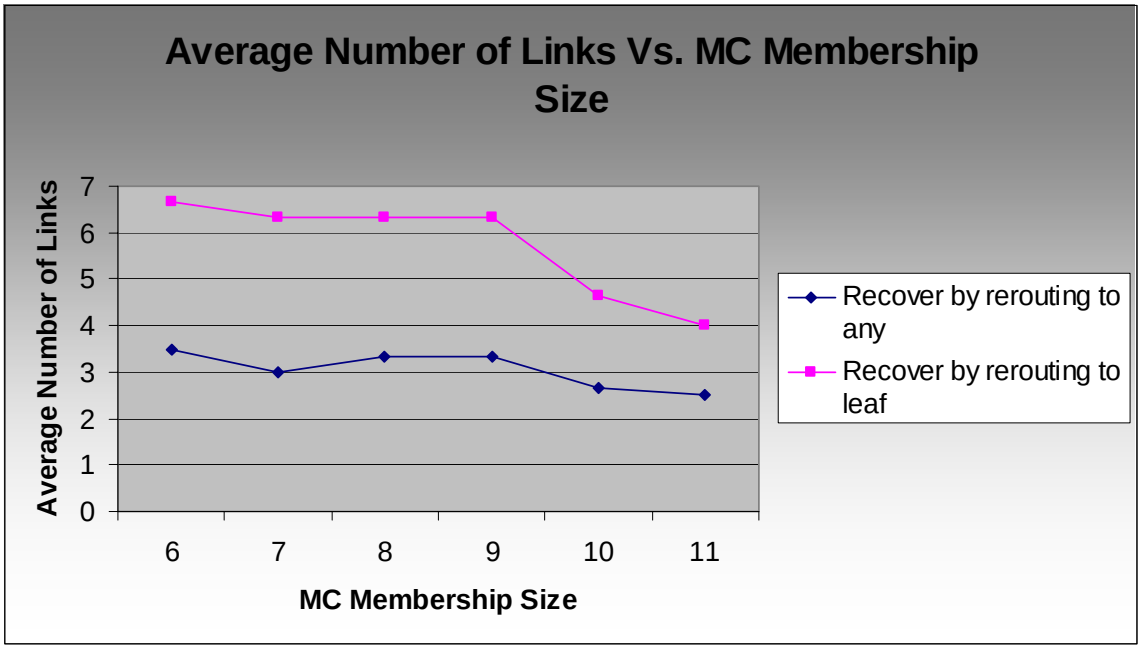

Figure 4.6 Average No. Of Links vs. MC Membership Size for Reactive Schemes

4.3.3 Results for Hybrid Schemes

As discussed earlier, to implement a hybrid scheme we use the parameter *threshold* to divide the primary multicast session into critical and non critical portions, which are to be handled by proactive and reactive recovery schemes, respectively. The plot in Figure 4.7 depicts the behavior of the proactive and reactive schemes with respect to the average number of links used for recovery in various combinations. Once again, P1, P2, R1, R2 in the plot, denote the protective and reactive schemes. **P1** refers to r*ecover by rerouting to neighbors*, **P2** refers to *recover by rerouting to source*, **R1** refers to *recover by rerouting to any*, and **R2** refers to *recover by rerouting to leaf*.

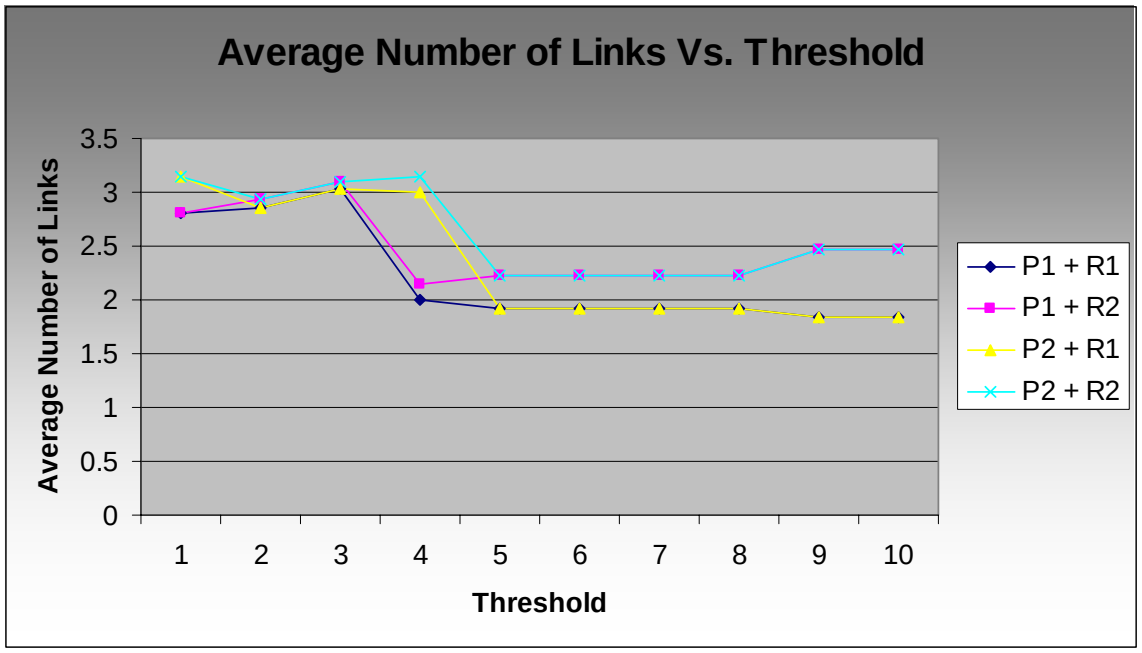

Figure 4.7 Average No. Of Links vs. MC Membership Size for Hybrid Schemes

As shown in Figure 4.7, the average number of links used for single link failure recovery goes down as the value of threshold is increased. The MC membership size is fixed at 11 in these simulations. A low threshold indicates more proactive protection is being provided than in the case of a higher threshold. In standalone mode P1 tends to use fewer links than P2 while R1 uses fewer links than R2 [See Section 4.3.1 and 4.3.2]. Thus, at lower thresholds P1 combined with R1 and R2 tend to use fewer links as compared to the schemes with P2. The behavior in combinations just confirms the results obtained in standalone modes for each of these methods. As the threshold value is increased we can see the effect of the reactive scheme setting in and the hybrid schemes using R1 tend to be better in terms of using fewer links than with schemes using R2.

4.4 Average Power Metrics

The objective of measuring this parameter is to quantify the quality of the signal upon recovery from a failure. This metric helps in comparing the various schemes from

the perspective of "*average power delivered*" to each of the destinations in the multicast session. In all the measurements, the power at the source is considered to be 1 and the power being delivered to each member in the multicast session is calculated based upon the formulations given in [5]. The formulations take into account the attenuation and splitting losses that an optical signal incurs while traveling on the fibers and through nodes. For a multicast session with a particular MC group size this metric indicates the average power delivered to the nodes upon recovering from different single link failures in that multicast session.

4.4.1 Results for Proactive Schemes

The behavior of this parameter is plotted in Figures 4.8 and 4.9. As can be seen, average power drops as the MC membership size increases indicating that as the size of the primary tree grows the average power delivered upon recovery decreases accordingly. The average power being delivered decreases as the number of destinations increases. The average power delivered by the scheme, *recover by rerouting to neighbors*, is higher than that of the scheme, *recover by rerouting to source*. This is because in *recover by rerouting to source* scheme there are higher chances of adding splits in the tree than with the *recover by rerouting to neighbors* scheme. Splitting causes more power to be lost in the recovered session which is reflected in the behavior obtained in the simulations. The plot in Figure 4.9 is obtained using the same network used for obtaining the plot in Figure 4.8 (i.e. USA Longhaul Network), but 6 links in the network are removed to reduce the connectivity while maintaining the 2-connectivity of the network.

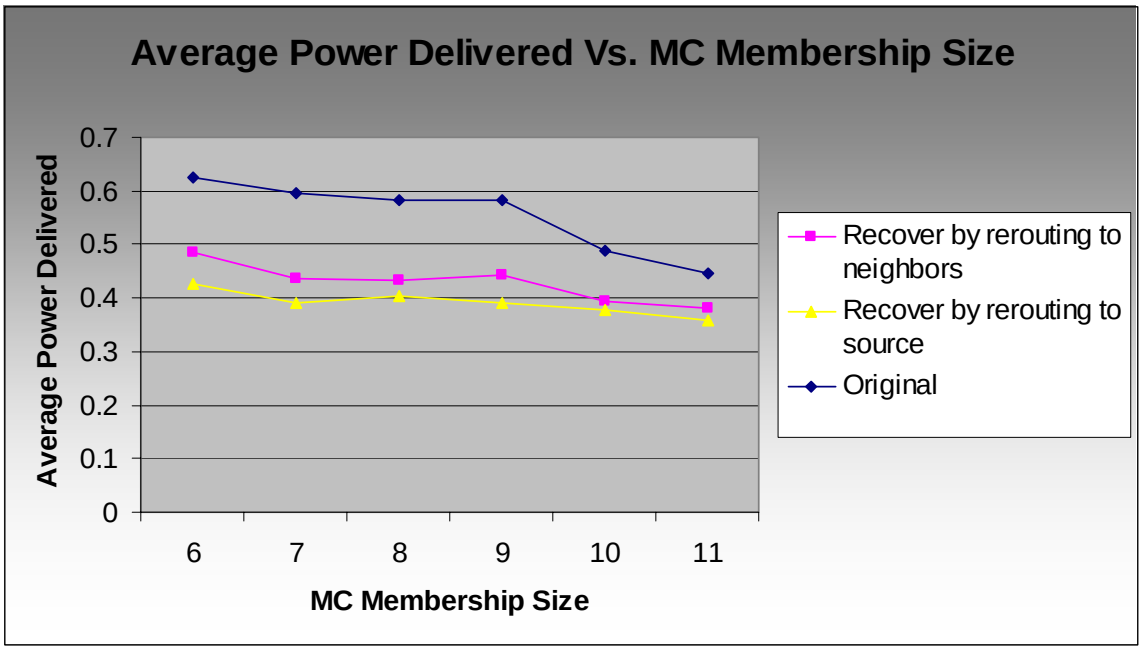

Figure 4.8 Average Power Delivered vs. MC Membership Size for Proactive Schemes

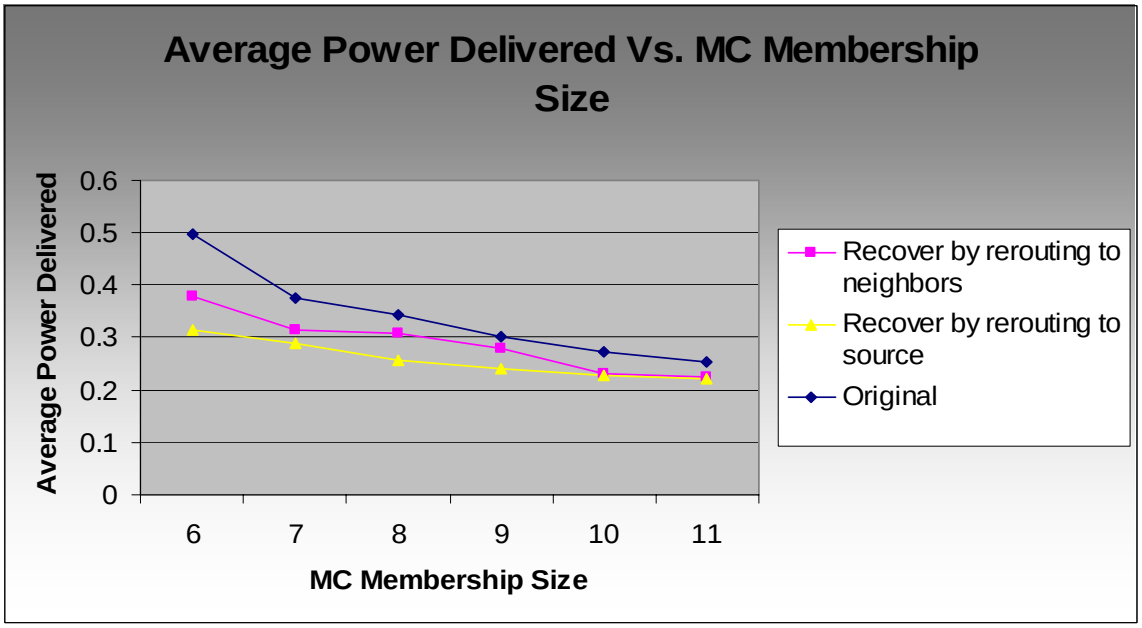

Figure 4.9 Average Power Delivered vs. MC Membership Size for Proactive Schemes (Different Connectivity)

4.4.2 Results for Reactive Schemes

Figures 4.10 through 4.12 show the average power delivered versus MC group size for the 3 reactive schemes. Different variants on the US Longhaul network with different connectivities were used to produce the results. The figures show that the average power delivered by the *recover by rerouting to leaf* scheme is less than that delivered by the *recover by rerouting to any* scheme. This might be because the average distance to leaf nodes tends to be larger than the rerouting distance in the latter scheme. The average power delivered in the MC tree tends to decrease as the MC membership size increases. As we can see in Figure 4.10, the variation in the average power is not significant in the initial portion and is also wavering. It is to be noted that each point on the plot is actually obtained by averaging out the average power delivered for at least 4 different single link failures. To ensure that this wavering behavior is actually due to the small number of samples used, we plotted the results once again by considering all possible single link failures in the multicast session one by one and then averaging out all results. The plots obtained by such an exhaustive sampling are shown in Figure 4.11. The variant of USA Longhaul network used for this plot had 7 links less than the original and still maintained 2-connectivity. The results shown in Figure 4.12 were obtained on a variant of the USA Longhaul network by removing 6 links from it while maintaining the 2-connectivity of the network. Here also, we see the trend in Figure 4.11 being repeated.

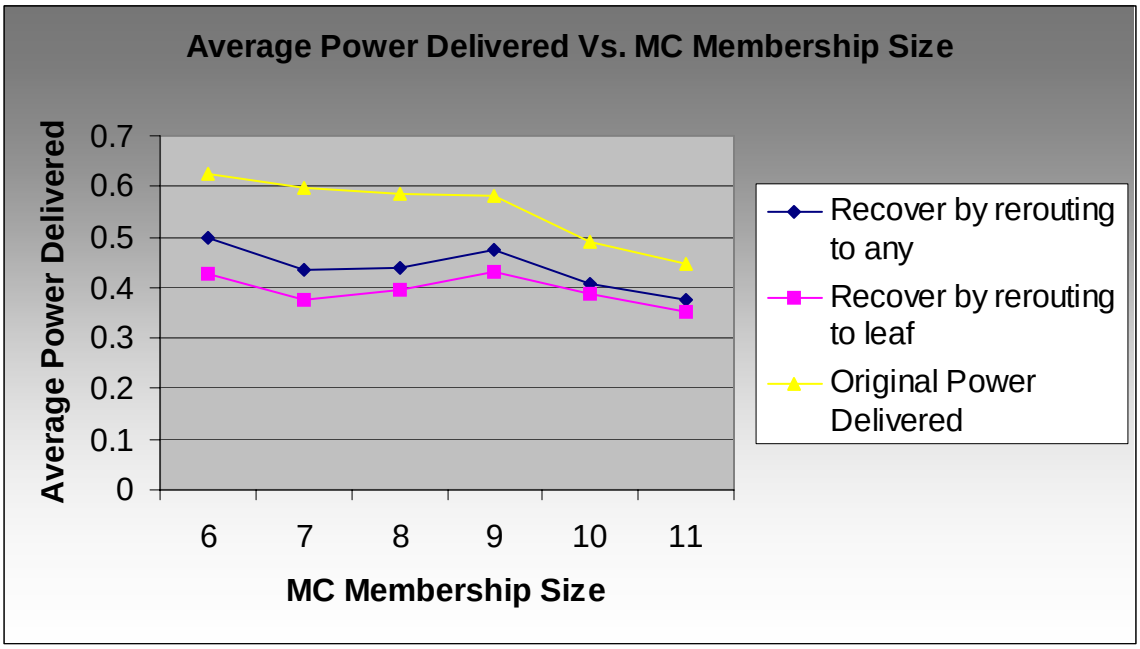

Figure 4.10 Average Power Delivered vs. MC Membership Size for Reactive Schemes

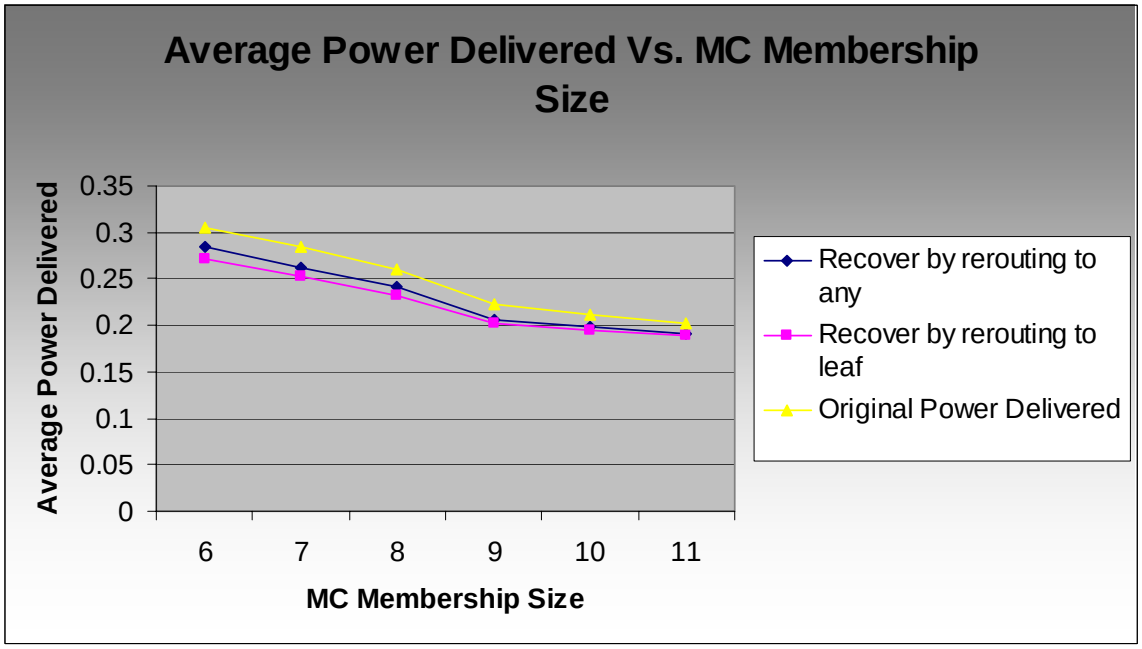

Figure 4.11 Average Power Delivered vs. MC Membership Size for Reactive Schemes (Different Connectivity)

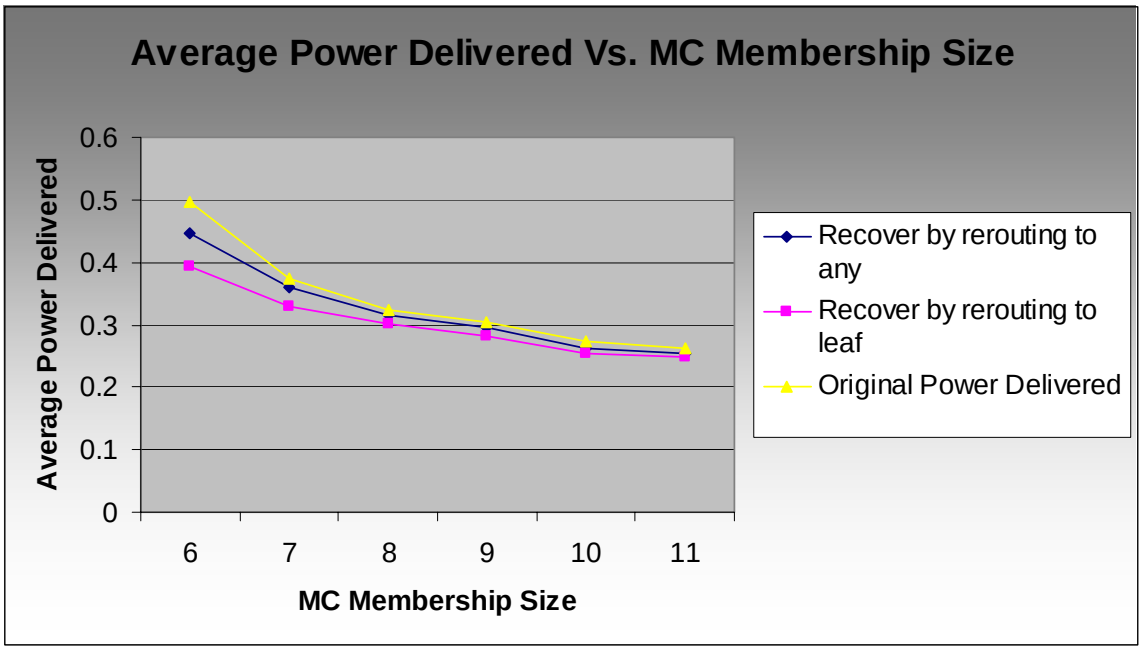

Figure 4.12 Average Power Delivered vs. MC Membership Size for Reactive Schemes (Different connectivity)

4.4.3 Results for Hybrid Schemes

The hybrid schemes have been studied in combinations and their behaviors have been plotted in Figures 4.13 and 4.14 for the USA Longhaul network with different connectivities. As can be observed the average power for single link failure recovery goes down as the threshold is increased. A low threshold value indicates that a lot of proactive protection is being provided than at a higher threshold value. The plot in Figure 4.14 was obtained using USA Longhaul network with 6 links removed while maintaining the 2-connectivity of the network. The MC membership size is fixed at 11 in these simulations.

In standalone mode P1 tends to deliver higher average power than P2 while R1 delivers higher average power than R2 [See Section 4.4.1 and 4.4.2]. At lower thresholds, the hybrid schemes with P1 deliver higher average power than the hybrid schemes with

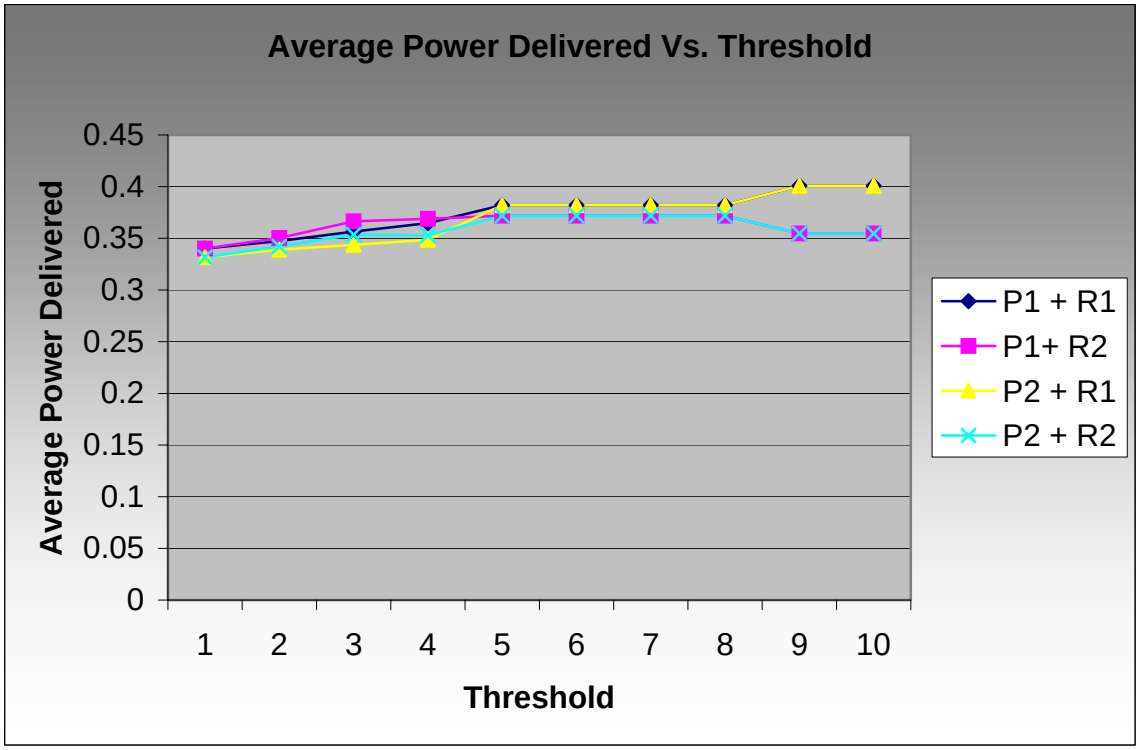

Figure 4.13 Average Power Delivered vs. MC Membership Size for Hybrid Schemes

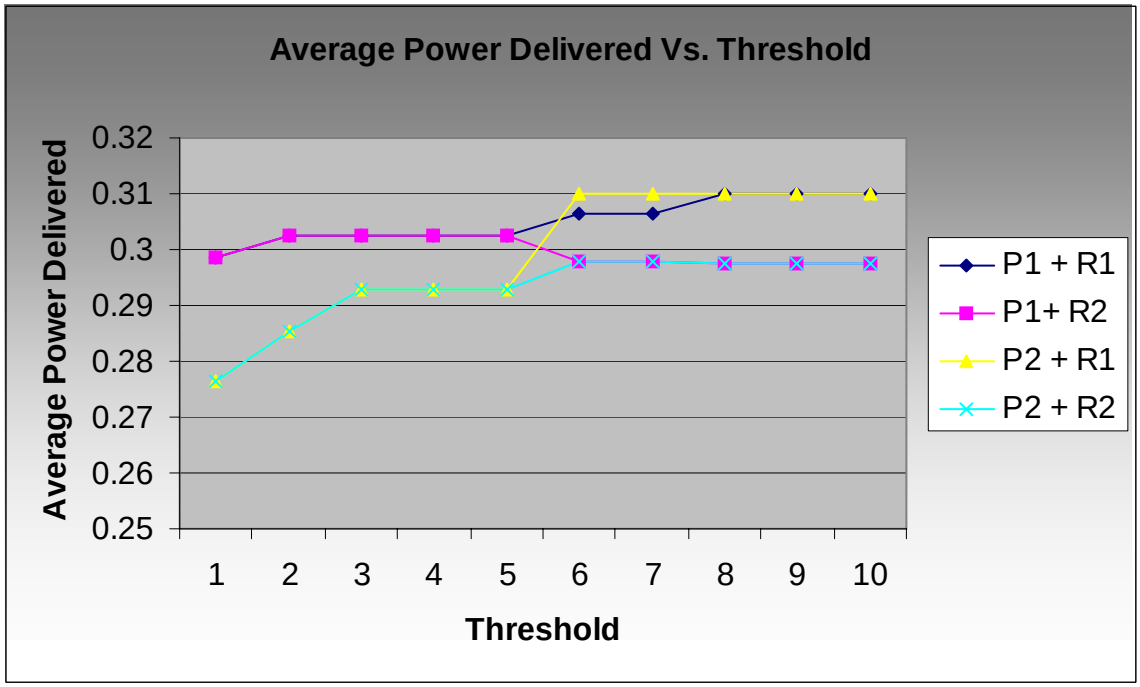

Figure 4.14 Average Power Delivered vs. MC Membership Size for Hybrid Schemes (Different Connectivity)

P2. As the threshold value is increased, we can see the effect of the reactive scheme setting in and the hybrid schemes using R1 deliver higher average power than with schemes with R2. The behavior in combinations just confirms the results obtained in standalone modes for each of these methods.

The plot in Figure 4.14 indicates the same behavior as that in Figure 4.13, but the differences in this case are more profound.

4.5 Percentage Change in the Average Power Delivered Metrics

This measure indicates the effectiveness of a recovery schemes in terms of power delivery by measuring the change in the average power delivered from the original average power that was delivered before the failure occurred. It reports the percentage change in the average power delivered in the recovered multicast session compared to the average power delivered in the original multicast session.

In all the measurements, the power at the source is considered to be unity and the power being delivered to each member in the multicast session is calculated based upon the formulations given in [5]. The formulations take into account the attenuation and splitting losses that an optical signal incurs while traveling in the network. The average power for any recovered session is obtained by averaging out the readings for at least 4 different single link failures.

4.5.1 Results for Proactive Schemes

Figures 4.15 and 4.16 depict the behavior of the proactive schemes. They report the percentage change in average power delivered from the original average power as the size of MC membership group is increased. The original power is indicated as a dark blue

line on the x-axis of the plot. We can note from the plots that the difference from the original average power delivered reduces as the size of the MC group increases.

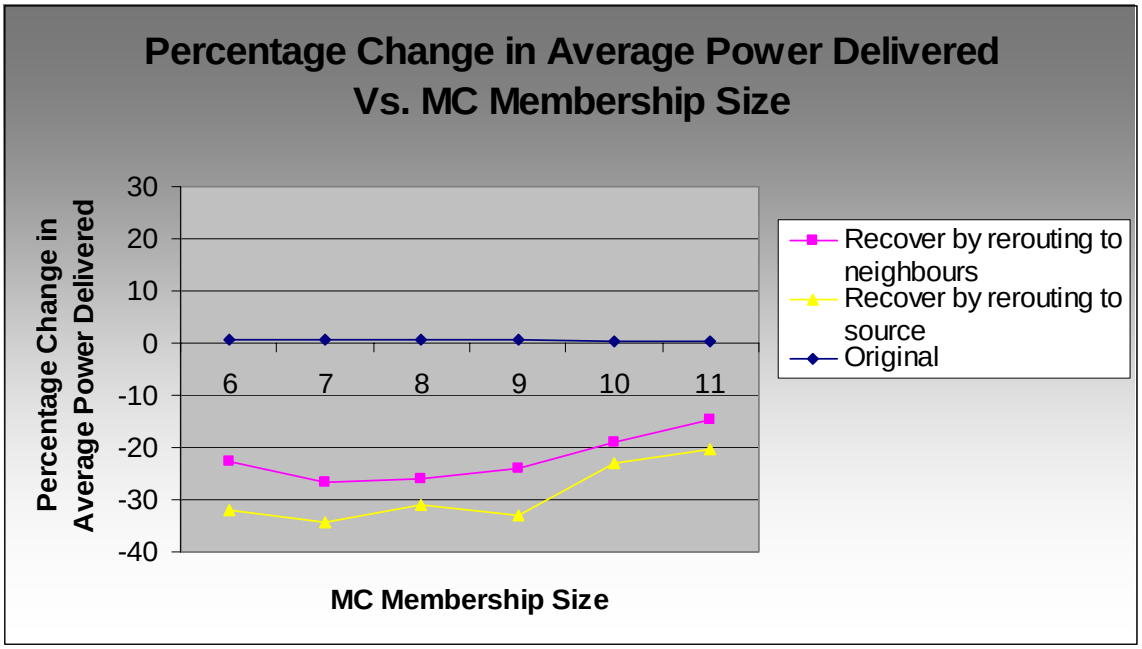

Figure 4.15 Percentage Change in Average Power Delivered Vs. MC Membership Size for Proactive Schemes

The reason for this behavior can be attributed to the fact that as the size of the MC group increases the number of useful nodes on the primary session that can be connected to in the recovery methods also increases. Thus, there are better chances of finding an alternate path that has a lower percentage of average power drop as compared to the original signal.

The percentage change follows an improvement pattern in the plots obtained in Figures 4.15, and 4.16. The plot in Figure 4.15 uses the original USA Longhaul network while the plot in Figure 4.16 uses a 2-connected variant with 6 links removed from the network

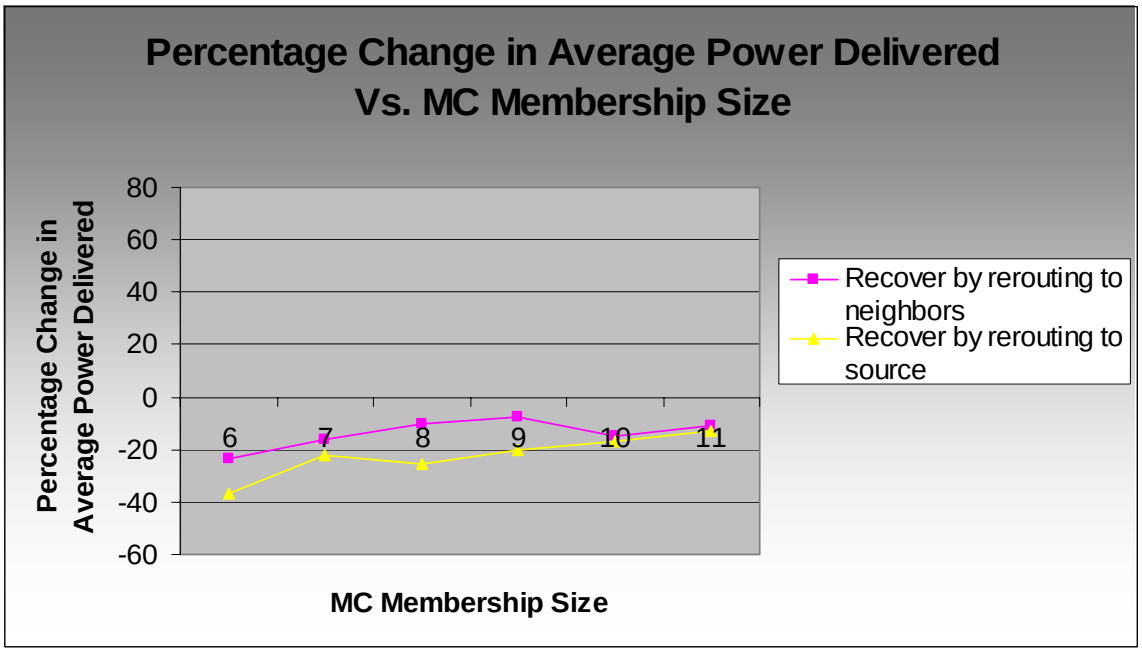

Figure 4.16 Percentage Change in Average Power Delivered Vs. MC Membership Size for Proactive Schemes (Different Connectivity)

4.5.2 Results for Reactive Schemes

Figures 4.17, 4.18, and 4.19 depict the percentage change in average power delivered from the original average power versus the size of MC group for the reactive schemes. The original power is indicated by the yellow line on the x-axis of the plots.

We can note from the plots that the difference from the original average power being delivered reduces as the size of the MC group increases. The reason for this behavior can be attributed to the fact that as the size of the MC membership group increases, the number of nodes on the primary multicast session to which alternate transmission path can be set increases. This effectively gives us an improvement in the percentage of average power drop from the original signal. The plots in Figures 4.18 and

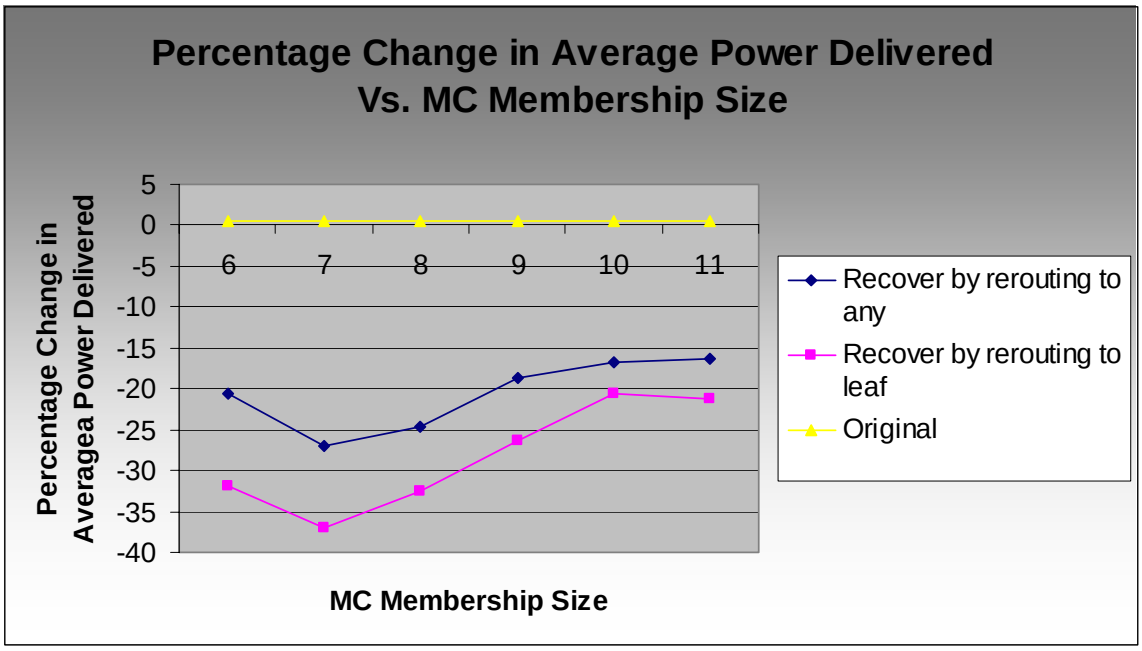

Figure 4.17 Percentage Change in Average Power Delivered Vs. MC Membership Size for Reactive Schemes

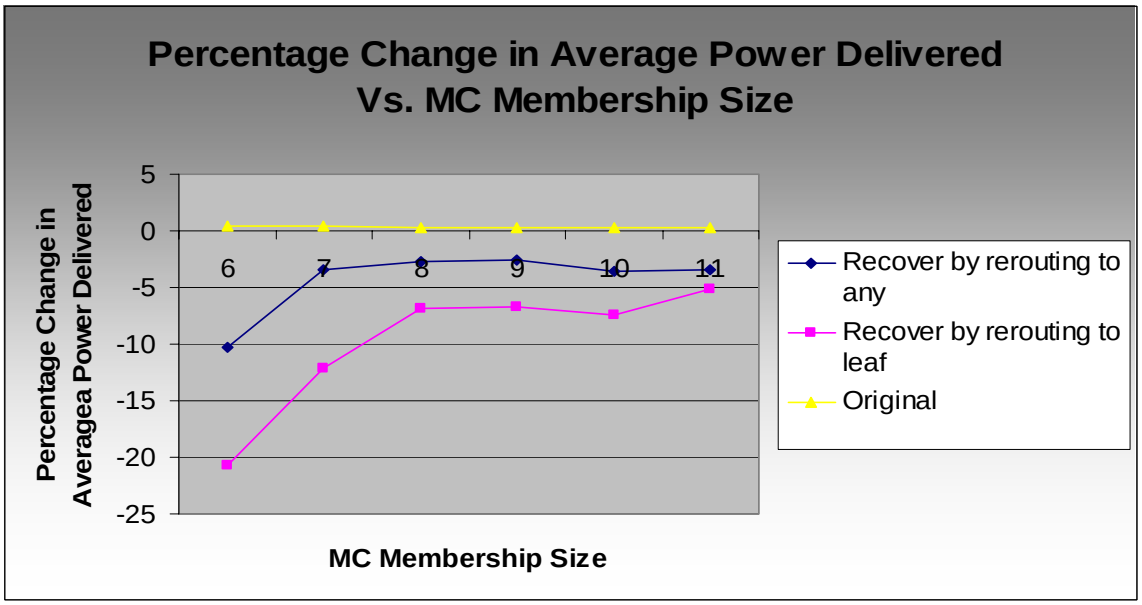

Figure 4.18 Percentage Change in Average Power Delivered Vs. MC Membership Size for Reactive Schemes (Different Connectivity)

and 4.19 have been obtained on the USA Longhaul network, by removing 6 and 7 links respectively from the network such that the bi-connectivity of the network is still

maintained. The plot in Figure 4.19 was obtained on a lesser connected network than the plot in Figure 4.18.

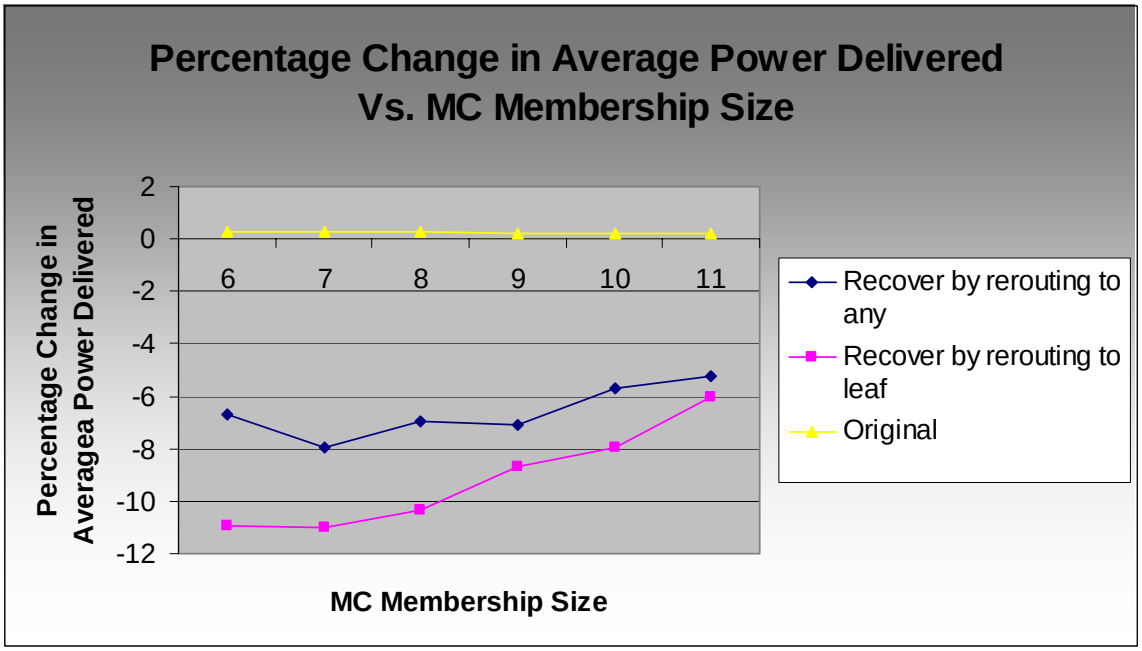

Figure 4.19 Percentage Change in Average Power Delivered Vs. MC Membership Size for Reactive Schemes (Different Connectivity)

4.6 3-Dimensional Plots for Network Usage and Power Efficiency

In the 3-Dimensional plots that we will discuss in this section, we try to study the network usage efficiency and the quality of the restoration schemes in terms of power delivery in the same plot. The network usage efficiency is measured by the average number of links used for recovering from a single link failure. The quality of restoration is measured in terms of the average power delivered in the recovered multicast session. The two parameters just mentioned have been measured for each of the schemes individually and against the MC membership size. The data for these 3-Dimensional plots has already been plotted in 2-Dimensional space as presented in Sections 4.3.1, 4.3.2,

4.4.1, and 4.4.2, respectively. The 3-D plots will help us to get a holistic view of the recovery solutions that we have provided in this thesis.

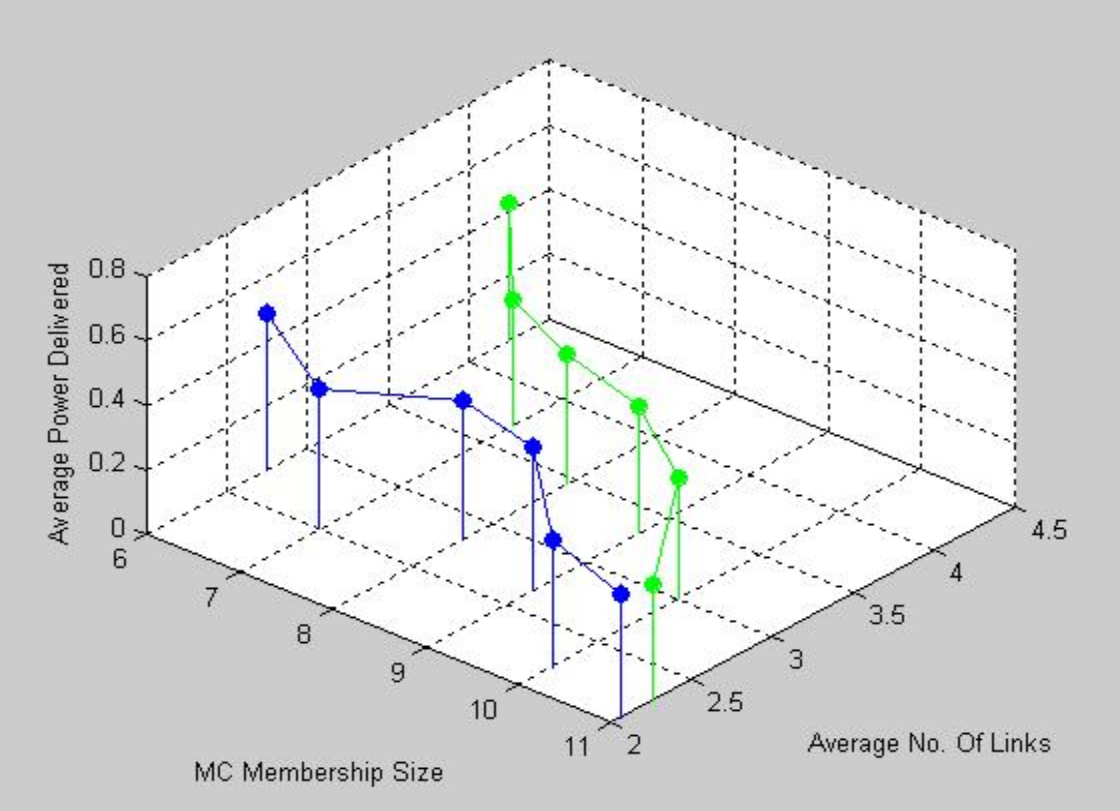

Figure 4.20 3-Dimensional Plot for Proactive Schemes

As mentioned in the earlier sections, the average power for a particular multicast session is calculated by averaging out the readings obtained for at least 4 individual single link failures. In a similar manner the average number of links used for recovering from a single link failure is actually the average of the results obtained for at least 4 individual single link failures.

The following color-coding is used for the various schemes shown in the plots:

Green: Recover by rerouting to source.

Blue: Recover by rerouting to neighbors.

Black: Recover by rerouting to leaf.

Magenta: Recover by rerouting to any.

The 3-Dimensional plot for the reactive schemes is depicted in Figure 4.20. For reactive schemes the 3-Dimensional plot is shown in Figure 4.21.

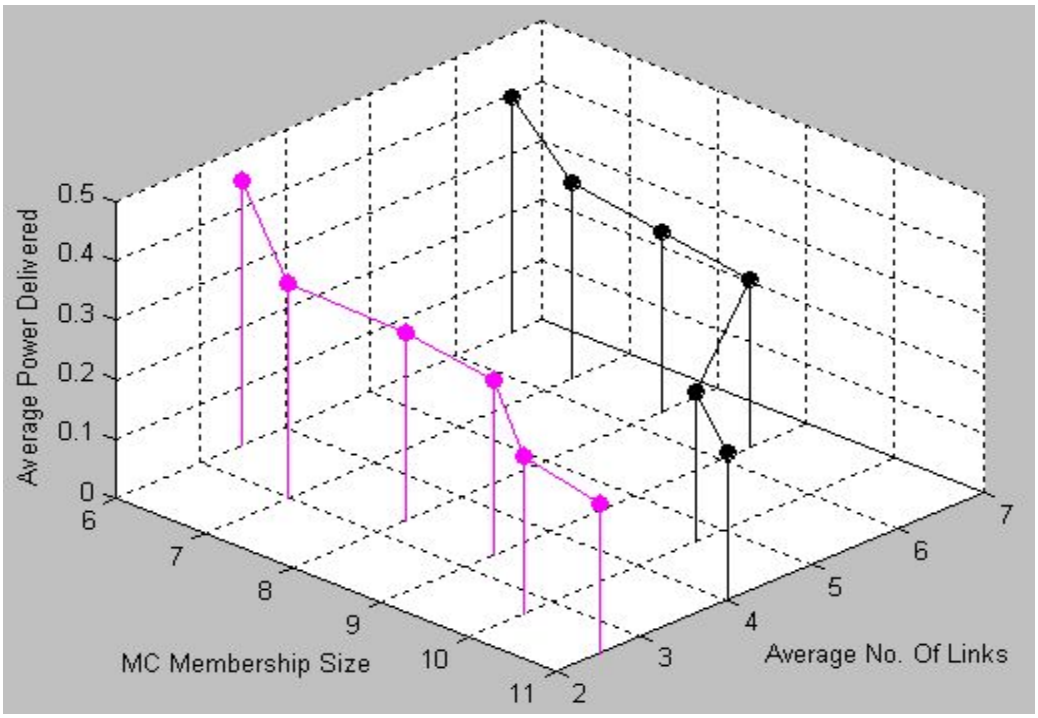

Figure 4.21 3-Dimensional Plot for Reactive Schemes

If we view the space in which the behavior is plotted in 3-D, then a scheme which falls in a region defined by small values for average number of links and high values for average power delivered over the entire range of MC group sizes is best. Such schemes will be efficient in using the resources while giving the best power quality upon recovery.

The 3-Dimensional plot shown in Figure 4.22 is the combination of plots for the reactive and proactive schemes. The schemes can be crudely ordered in the following order in terms of overall performance: *recover by rerouting to neighbors*, *recover by rerouting to any*, *recover by rerouting to source*, and *recover by rerouting to leaf.*

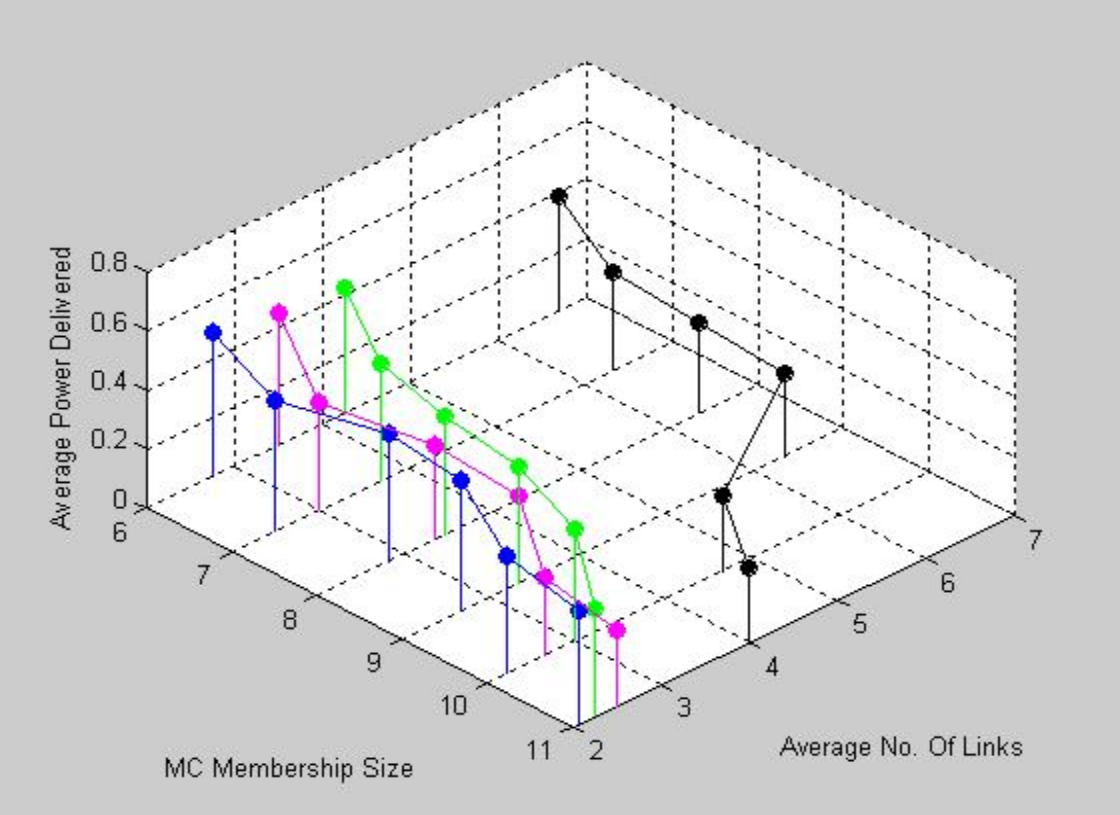

Figure 4.22 3-Dimensional Plot of All Schemes

Chapter 5

Design Specifications of the Simulator

5.1 Overview

This chapter gives the high level and low level design view of the simulator. All simulations for the protection and reaction heuristics have been developed using Java as the development language. The simulator has been totally modularized, so as to enable the user to run any combination of protective and reactive heuristics. Among the three protection heuristics, the "*Recover by re-computation*" is a standalone approach to the problem. The other two protective schemes and the two reactive schemes give us a total of eight possibilities i.e. 4 in combinations and 4 in standalone modes by themselves. Please refer to the previous chapter for a description on how the proactive and reactive schemes can be used in combinations.

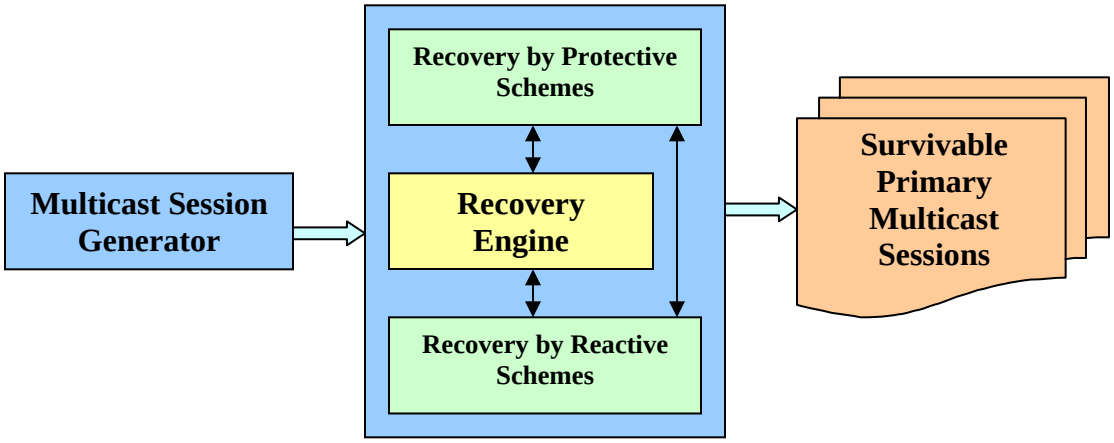

Figure 5.1 Simulator Modules

The system mainly comprises of two subsystems/units:

a. **Multicast Session Generator:** The task of this unit is to generate primary sessions (forest), which will be acted upon in the next stage to make them survivable. This unit

is based upon a heuristic approach proposed in [1] for the design of power efficient multicast forests.

b. **Recovery Engine:** This unit acts upon the multicast forest generated by the session generator to make the primary multicast sessions survivable. It acts on the basis of a reaction, a protection, or a hybrid scheme. The recovery engine has been designed for using any of the proactive schemes (except for "*Recover by recomputation*") with any of the reactive schemes.

5.2 Implementation Elements

The simulator has been designed following an Object-Oriented design methodology and using a UML (Unified Modeling Language) toolkit. All these classes share a common namespace, *com.opticalnetworks.multicast.survive*. The subsystem, **Multicast Session Generator** has been implemented by the classes *MCTree*, *GraphNode*, *Edge*, *MCSession* and *ShortestPaths*. The subsystem **Recovery Engine** has been implemented by the classes *Segments*, *ProtectedSession*, *ProtectNetworkElements*, and interface *ISurvive*.

5.3 Class Design Specifications

The class design was carried out using UML. Each of the representations is a UML notation for it. Further information can be obtained from the detailed *javadoc* style documentation accompanying the simulator. Class descriptions are summarized next:

ISurvive

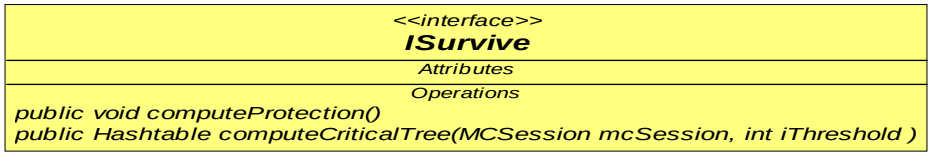

Figure 5.2 Interface ISurvive

This interface forces the algorithms, based on the concept of critical trees, to implement the two methods declared within the interface. Algorithms based on the concept of critical tree are thus required to implement this interface and be of type *ISurvive*.

GraphNode

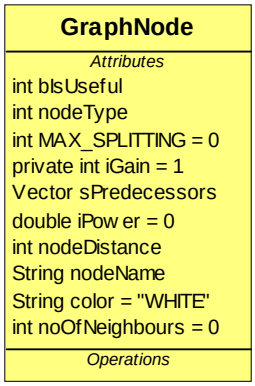

Figure 5.3 Class GraphNode

This class stores the details and attributes of a single vertex of the network.

Segments

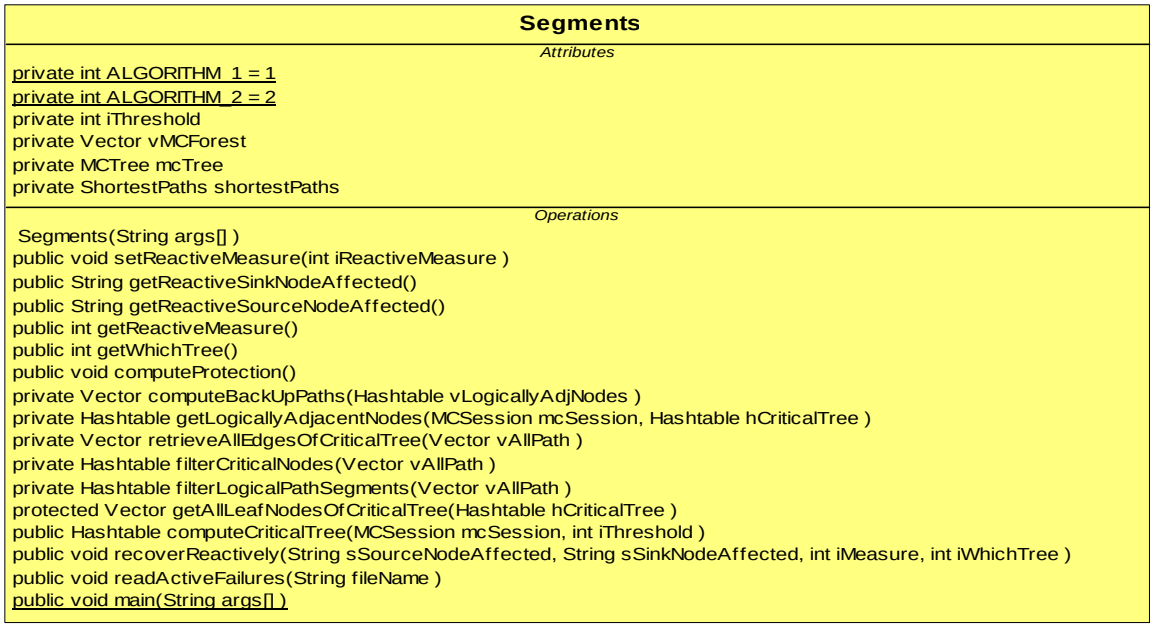

Figure 5.4 Class Segments

The protection algorithms *recover by rerouting to the neighbors* and *recover by rerouting to source* and the reactive algorithms, *recover by rerouting to any* and *recover by rerouting to leaf* are realized in this class. Thus, the class gives us the flexibility of using any of the proactive schemes with any of the reactive schemes, resulting in a total of four combinations. This class is of type *ISurvive*.

Edge

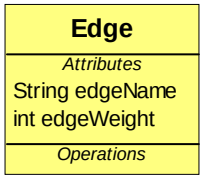

Figure 5.5 Class Edge

This class stores the details of a single edge of the network.

MCSession

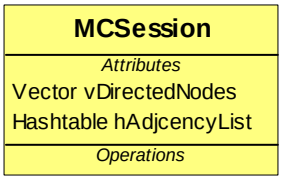

Figure 5.6 Class MCSession

This class stores the node list and the adjacency list for a multicast session. One or several of these are stored as a collection in the class *MCTree* to represent a multicast forest for a given multicast group.

ProtectedSession

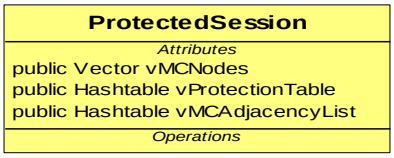

Figure 5.7 Class ProtectedSession

This class stores the protection information for a primary multicast session. The hashtable *vProtectionTable* is indexed by network element name (in this case an edge). It stores the alternate multicast session, (in case of that element's failure) which itself could be a multicast forest.

ProtectNetworkElements

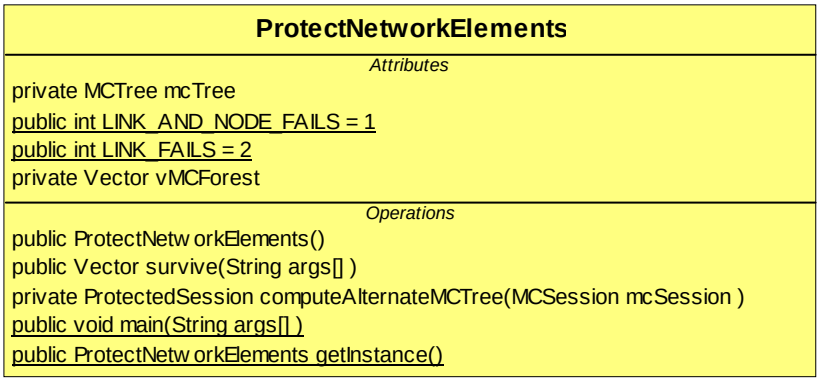

Figure 5.8 Class ProtectNetworkElements

This class fails each of the edges present in the primary session and re-computes the alternate multicast tree for the affected destinations. The functionality for the creation of the alternate tree comes from the class *MCTree*. All of these alternate multicast trees are stored in a *hashtable* which is indexed by the name of the edge that can fail.

ShortestPaths

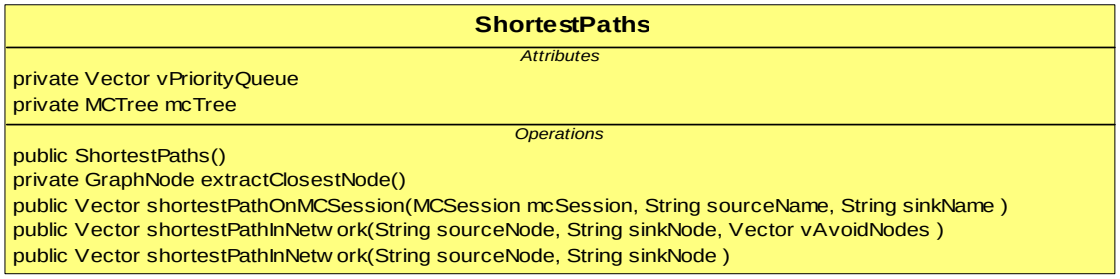

Figure 5.9 Class ShortestPaths

This class embodies a generic functionality needed by the classes discussed above. It provides the functions to compute shortest paths between the nodes.

MCTree

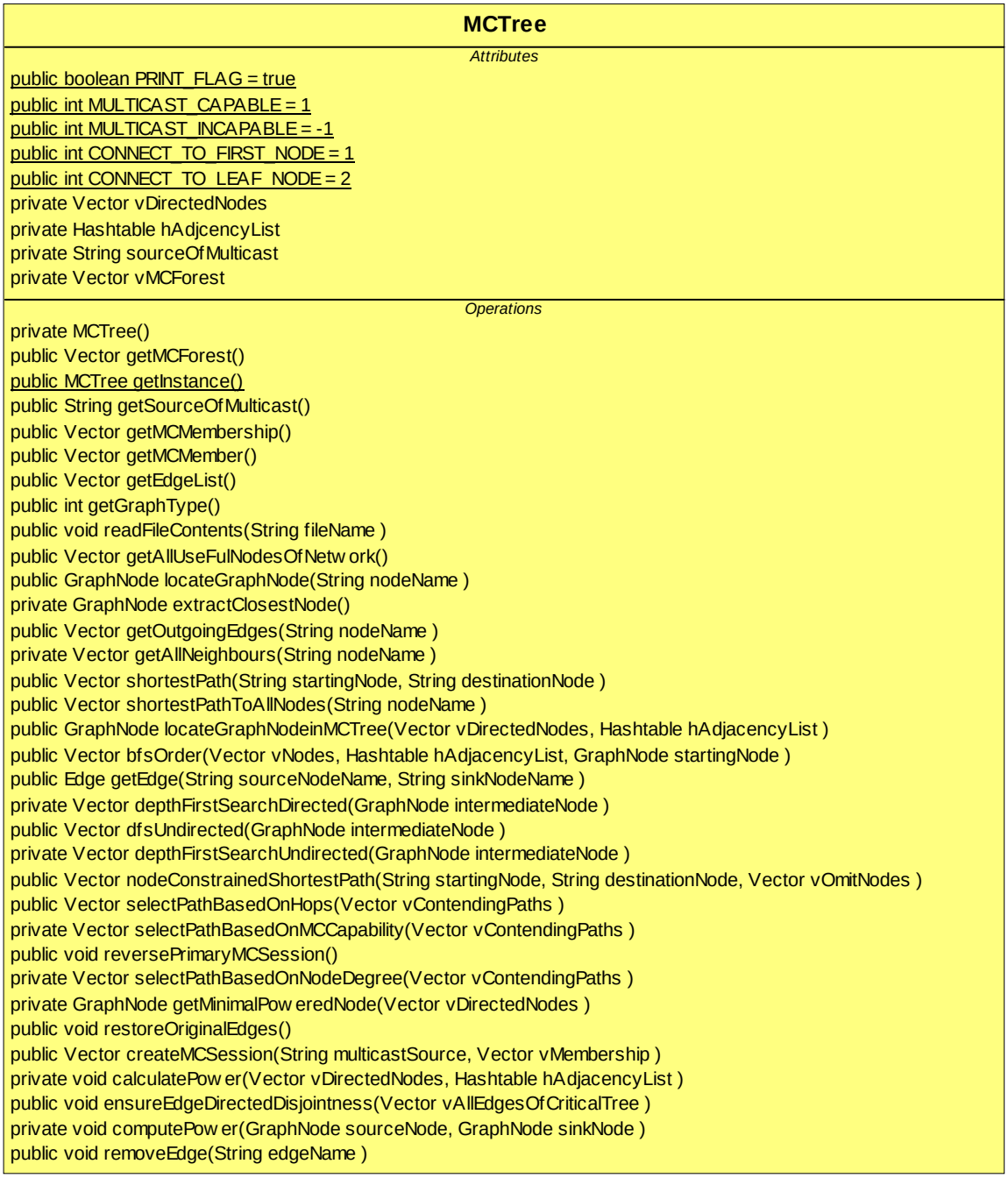

Figure 5.10 Class MCTree

MCTree is a singleton class. This class is responsible for the generation of primary multicast sessions which are made survivable by the classes *Segments* and *ProtectNetworkElements*. The number of multicast sessions for a given multicast

membership group can be more than 1. All the multicast sessions are stored as instances of the class *MCSession* in a collection. The edges and nodes of the network are stored in this class's instance only. Since this class is a singleton class it is ensured that we are maintaining a single state for the network in the whole simulation environment.

The class interaction diagram showing the relationship among the various classes is shown in Figure 5.11 below:

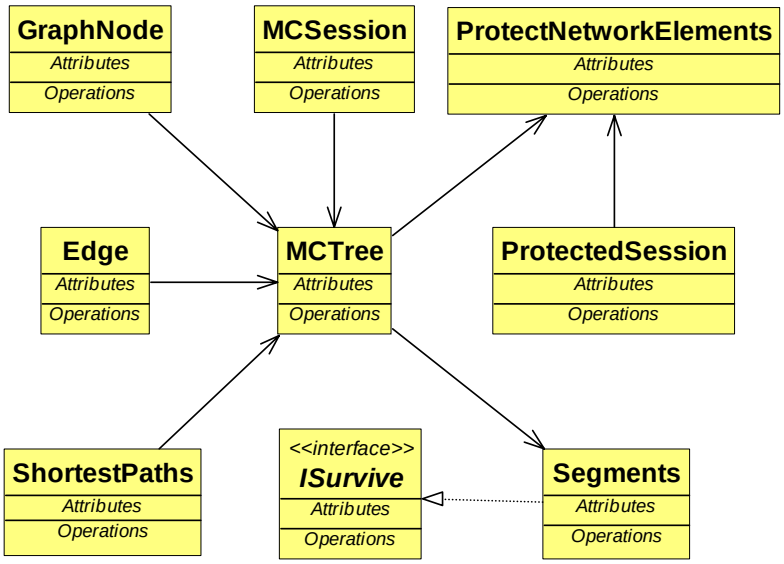

Figure 5.11 Class Interaction Diagram

The association between the class *GraphNode* and *MCTree* is one to many. Multiple copies of the class *GraphNode* each encapsulating the details of a node of the network are persisted in the class *MCTree*. The edges of the network are treated by a similar relationship using the one-to-many relation between the classes *Edge* and *MCTree*. The class *Segments* realizes the interface *ISurvive*. All the classes that build upon the idea of critical tree are required to be of type *ISurvive*. Each of the associations between *MCTree* and *ProtectNetworkElements*, *ProtectedSession* and *ProtectNetworkElements*, and *ShortestPaths* and *MCTree*, is one-to-one.

5.4 Executing the Simulator

To use the command prompt to run the simulator, we navigate to the directory where the Java classes are kept. We run the following command template and set the parameters depending upon the choice of protection or reaction scheme or the level of protection and reaction scheme in a hybrid approach. If the simulator is being run from an integrated development environment (IDE), then in that case, all these parameters can be set within the environment of the IDE.

[Drive]:directory\>java com.opticalnetwork.multicast.survive.Segments <input_network> *threshold protective_scheme* <active_failures_file> *reactive_scheme*

Where, <input_network> is the path of the flat file describing the optical network,

threshold is the value of the critical threshold,

protective_scheme indicates which of the protective schemes to use. A value of 1 is used for *"Recover by rerouting to neighbors"* and value of 2 is used for *"Recover by rerouting to source"*. This command does not execute the heuristic "*Recover by re-computation"*, whose instructions follow this description.

<active_failures_file> is the path of the flat file describing the failure that will be handled using the reactive scheme.

reactive_scheme indicates which of the reactive schemes to use. A value of 1 is used for *"Recover by rerouting to any"* and value of 2 is used for *"Recover by rerouting to leaf"*.

For example, to provide protection by "*Recover by rerouting to neighbors*" and reactive recovery by "*Recover by rerouting to any*" with a threshold of 3, we would specify the following command on the prompt:

[Drive]:directory\>java com.opticalnetwork.multicast.survive.Segments

<input_graph_file> 3 1 <active_failure_details_file> 1

To execute the heuristic "*Recover by re-computation*", the following command template needs to be used:

[Drive]:directory\>java com.opticalnetworks.multicast.survive.ProtectNetworkElements

<input_network>

Where, <input_network> is the path of the flat file describing the optical network.

5.5 Format of Input Files

The network is described in a flat file in the following format.

Directed_Flag/Undirected_Flag

No_Of_Nodes

MC_Source

MC_Membership_List

*Name Type NoOfNeighbors (Name_Of_Neighbor Weight_of_the_node)**

Where * indicates zero or more occurrences of the pattern. The value of Directed_Flag/Undirected_Flag is kept at 1, to indicate an undirected graph. The field No_Of_Nodes indicates the total number of nodes in the network. MC_Source indicates the source of the multicast session. The MC_Membership_List specifies the destination nodes of the multicast session. Following this there must be as many rows as specified by the value of No_Of_Nodes. Each row specifies the name, type and the number of its neighbors. The pattern (Name_Of_Neighbor Weight_of_the_Edge)* will be repeated as many times as given by the number NoOfNeighbors.

An example of the network input file follows:

0
14
14
12 8 11 5 6
1 1 3 2 1 3 1 4 1
2 1 3 1 1 3 1 6 1
3 1 3 1 1 2 1 8 1
4 1 3 1 1 10 1 5 1
5 1 3 4 1 7 1 6 22
6 1 4 2 1 5 1 13 1 9 1
7 0 2 5 1 8 1
8 1 3 7 1 3 1 11 1
9 0 2 6 25 11 1
10 1 3 4 1 14 1 12 1
11 1 4 8 1 12 1 14 1 9 1
12 1 3 10 1 11 1 13 1
13 1 3 6 1 12 1 14 1
14 1 3 11 1 13 1 10 1

This structure defines the NSF network, the source of the multicast session is 14 and the destinations are 12, 8, 11, 5, and 6. All the edges have been given a weight of unity. The reactive failure that is to occur once the protection is provided is described in the following format in a flat file:

Link_To_Be_Failed
WhichPrimaryTree

An example of the reactive failure file is:

23-25
0

The above indicates that the link 23-25 fails in the multicast session indexed by element 0 in the forest. Since there could be multiple trees in the forest, so, the field *WhichPrimaryTree* tells us which particular tree to look for in the forest.

5.6 Simulation Environment

Hardware: Intel Pentium 4 machine with clock speed of 2.2 GHz and 1 GB RAM.

Software: Java(TM) 2 SDK1.4.2 was used on Windows 2000 operating system.

Tools Used: GraphViz 1.4.2 for graph illustrations, DescribeEnterprise 3.0 for UML modeling.

Chapter 6

Conclusions and Future Directions

This thesis presented heuristic schemes to provide recovery from single link failures in WDM optical networks. Single link failures are important causes of network impairments. This work can be viewed as a follow up on the earlier work on building power efficient multicast trees, reported in [1]. We have attempted to make those power-efficient multicast trees, resilient to single link failures. We presented five heuristic schemes for recovering from single link failures. Three of these are classified as protective methods and the other two are reactive methods. The protective schemes will be activated at the time of creation of the primary multicast session. The reactive schemes are applied for recovery when the failure actually occurs. Since each approach has its own advantages and disadvantages, we have proposed a novel method to amalgamate the two schemes in the same primary multicast session. We have introduced the novel concepts of threshold and critical tree to demarcate the primary session into critical and non-critical portions. The critical portion is serviced protectively whereas the rest of the tree is serviced reactively for single link failures. The primary objective in all the recovery methods has been to recover from the failure. The performance of these five algorithms under several constraints, in standalone and hybrid combination modes, has been studied.

The metrics used to evaluate the schemes included average power delivered to the destinations, percentage change in power delivered to the destinations, number of links used for recovery and the time taken to compute the alternate backup paths. The recover by recomputation scheme uses more links for recovery and takes more time to execute as

compared to other schemes. The rest of the schemes can be crudely ordered in the following order of performance in terms of network usage and power efficiency: recover by rerouting to neighbors, recover by rerouting to any, recover by rerouting to source, and recover by rerouting to leaf.

While we have considered only recovery in this work, future work can incorporate other parameters and objectives like time complexity, power efficiency in the recovery algorithms. In the current work we have limited the failures to just a single link failure. Future research can scale it further to recover from multiple link and node failures.

References

[1]. K. Buddharaju, “Design of Power Efficient Multicast Algorithms for Sparse Split WDM Networks”, A Master’s thesis, Dept. of Electrical Engineering, Louisiana State University, Baton Rouge, 2003.

[2]. N. K. Singhal and B. Mukherjee, “Protecting Multicast Sessions in WDM Optical Mesh Networks”, Journal of Lightwave Technology, Vol. 21, No. 4, April 2003.

[3]. S. Ramamurthy and B. Mukherjee, “Survivable WDM mesh networks, Part I-Protection”, Communications, 1999. ICC '99. IEEE International Conference on, 1999, pp. 2023 –2030, vol.3, 1999.

[4]. G. Mohan and C. Siva Ram Murthy, “Lightpath Restoration in WDM Optical Networks,” IEEE Network Magazine, vol. 12, no. 6, pp. 24-32, November/December 2000.

[5]. Y. Xin and G. Rouskas, “Light Tree Routing Under Optical Power Budget Constraints,” OSA Journal of Optical Networking (JON), vol. 3, no. 5, pp. 282-302, May 2004.

[6]. C. Murthy and M. Guru Swamy, “WDM Optical Networks”, Prentice Hall PTR.

[7]. B. Mukherjee, Optical Communication Networks, McGraw-Hill, New York, NY, 1997.

[8]. Cisco Systems, “Introduction to DWDM for Metropolitan Area Networks”, http://www.cisco.com/univercd/cc/td/doc/product/mels/dwdm/index.htm, 1999.

[9]. http://www.commspecial.com/fiberguide.htm.

[10]. http://www.iec.org/online/tutorials/acrobat/fiber_optic.pdf.

[11]. http://www.telebyteusa.com/foprimer/foch2.htm.

[12]. Wavelength Division Multiplexing – WDM by J. Hecht “Understanding Fiber Optics”, Prentice Hall PTR.

[13]. http://www2.rad.com/networks/1999/wdm/wdm.htm

[14]. http://www.cis.ohio-state.edu/~jain/cis788-99/ftp/dwdm/index.html

[15]. http://www2.rad.com/networks/1999/wdm/wdm.htm

[16]. K. B. Kumar. and J. M. Jaffe, “Routing to Multiple Destinations in Computer Networks,” IEEE Transactions Commun., Vol.COM-31, no. 3, Pg. 343-351, Mar. 1983.

[17]. http://searchnetworking.techtarget.com/sDefinition/sid7_gci214539,00.html.

[18]. http://www2.rad.com/networks/1999/wdm/wdm.htm.

[19]. http://www.iec.org/online/tutorials/opt_net/topic11.htm.

[20]. C. H. Lin and C. H. Wang, “A Hybrid Multicast Scheduling Algorithm for Single Hop WDM Networks.”

[21]. Z. Oritz, G. Rouskas, and H. G. Perros, “Scheduling of Multicast traffic in tunable receiver WDM networks with non negligible tuning latencies,” In Proceedings of SIGCOMM `97, Pg 301-310. ACM, September 1997.

[22]. Z. Oritz, G. Rouskas, and H. G. Perros, “Maximizing Multicast throughput in WDM networks with tuning latencies using the virtual receiver concept,” European Transactions on Telecommunications, 11(1): 63-72, January/February 2000.

[23]. WDM-Based All-Optical Networks for Future Internet Applications M. Mukunda Rao and G. De Marchis Optical Communication Division Fondazione Ugo Bordoni (Settore 2) Via Baldassare Castiglione 59 00142 Rome / Italy.

[24]. G. Rouskas, and H. G. Perros, “A Tutorial on Optical Networks,” Networking 2002 Tutorials, LNCS 2497, Pg. 155–193, 2002.

[25]. M. Medard, S. G. Finn, R. A. Barry, and R. G. Gallager, “Redundant Trees for Preplanned Recovery in Arbitrary Vertex-Redundant or Edge-Redundant Graphs”, IEEE/ACM Transactions on Networking, Vol. 7, No. 5, October 1999.

[26]. X. Zhang, J. Wei and C. Quiao , “Constrained Multicast Routing in WDM Networks with Sparse Light Splitting.”

[27]. K. Lee, Kai-Yueng Siu, “An Algorithmic Framework for Protection Switching in WDM Networks”.

[28]. R. Bhandari, “Survivable Networks - Disjoint Path Algorithms”, AT&T Laboratories.

[29]. R. Bhandari, “Survivable Networks – Algorithms for Diverse Routing”, AT&T Laboratories, Kluwer Publications.

[30]. G. Kithlanagamangala., “Efficient Embedding of Virtual Hypercubes in Irregular WDM Optical Networks”, A Master’s thesis, Dept. of Electrical Engineering, Louisiana State University, Baton Rouge, 2003.

[31]. R. Malli, X. Zhang, and C. Qiao. Benefits of multicasting in all-optical networks. In SPIE Proceedings, All Optical Networking, pages 209–220, November 1998.

[32]. M. Ali and J. S. Deogun, “Cost Effective Implementation of Multicast Wavelength Routed Networks,” 0733-8724 2000 IEEE.

Vita

Sateesh Chandra Shekhar received the degree of Bachelor of Engineering (Computer Engineering) from Pune University, Pune, India, in 1999, and will recieve the degree of Master of Science in August 2004, in Electrical Engineering, from Louisiana State University, Baton Rouge. He was a silver medalist for his outstanding academic performance in the undergraduate studies. From 1999 to 2003, he worked with Infosys Technologies Ltd., and Sasken Communication Technologies Ltd. as a Senior Software Engineer. In 2003, he co-founded Ideavate Solutions Inc. in New Jersey (http://www.ideavate.com). Currently, he is the Director of Technology Solutions Group and also spearheads the corporate quality initiatives at Ideavate Solutions Inc.