# RESILIENT OPTICAL MULTICASTING
# UTILIZING CYCLES 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**
**Abhishek Satyendranath**
**Bachelor of Engineering (Telecommunication Engineering)**
**Visveswaraiah Technological University, 2002**
**August 2006**

***To the most important people in my world, my parents, Dr. Satyendranath and Mrs. Pushpalatha Satyendranath, for your unconditional love and support.***

# Acknowledgements

***“At times our own light goes out and is rekindled by a spark from another person. Each of us has cause to think with deep gratitude of those who have lighted the flame within us.”***

***-Albert Schweitzer***

I owe my debt of gratitude to my advisor Dr. Ahmed El Amawy for his academic guidance, encouragement, and above all patience. My sincere appreciation and thanks to him for taking time off his busy schedule to help me during the final stages of the thesis. I am also grateful to Dr Jerry L Trahan and Dr. Hsiao-Chun Wu for agreeing to serve on my committee.

I am absolutely certain this work would never have seen the light of day without the support of Dr. Fred A Rainey of the Biological Sciences Department. He has been more of a friend to me than a boss.

To my parents I owe the very reason for my existence. Many thanks to them for instilling in me values of hard work and persistence and encouraging me to always dream big. My special thanks to Sreetal who encouraged me to come to the US to realize my dreams.

I would like to thank all my colleagues and friends here at LSU, who have helped make my stay here, a very pleasant and memorable one. My special thanks to Varma Indukuri for helping me out with my simulator.

My thanks to the entire optical research community who have, directly or indirectly, influenced my thinking and research.

# Table of Contents

# List of Figures

# Abstract

High capacity telecommunications of today is possible only because of the presence of optical networks. At the heart of an optical network is an optical fiber whose data carrying capabilities are unparalleled. Multicasting is a form of communication in wavelength division multiplexed (WDM) networks that involves one source and multiple destinations. Light trees, which employ light splitting at various nodes, are used to deliver data to multiple destinations. A fiber cut has been estimated to occur, on an average, once every four days by TEN, a pan-European carrier network. This thesis presents algorithms to make multicast sessions survivable against component failures. We consider multiple link failures and node failures in this work.

The two algorithms presented in this thesis use a hybrid approach which is a combination of proactive and reactive approaches to recover from failures. We introduce the novel concept of *minimal-hop cycles* to tolerate simultaneous multiple link failures in a multicast session. While the first algorithm deals only with multiple link failures, the second algorithm considers the case of node failure and a link failure. Two different versions of the first algorithm have been implemented to thoroughly understand its behavior. Both algorithms were studied through simulators on two different networks, the USA Longhaul network and the NSF network.

The input multicast sessions to all our algorithms were generated from power efficient multicast algorithms that make sure the power in the receiving nodes are at acceptable levels. The parameters used to evaluate the performance of our algorithms

include computation times, network usage and power efficiency. Two new parameters, namely, recovery times and recovery success probability, have been introduced in this work. To our knowledge, this work is the first to introduce the concept of minimal hop cycles to recover from simultaneous multiple link failures in a multicast session in optical networks.

# Chapter 1

# WDM Optical Networks

Since the early 1980's the evolution of optical networks has revolutionized the telecommunication industry. The use of optical fibers as a data transmission medium has not only met the demands for very high bandwidth requirements but also given rise to more secure and reliable communications. The capacities of fiber optics are simply amazing, and optical fibers perform far better than their electrical counterparts. The BER (Bit Error Rate) for a typical optical fiber is in the range of $10^{-13}$ to $10^{-14}$ whereas for a copper cable it can be anywhere between $10^{-3}$ and $10^{-4}$ [2]. The cost per foot for fiber is around 20 cents compared to 13 cents for copper. Optical fibers are easier to install due to lower weight and thickness. Fibers are immune to RFI (Radio Frequency Interference) and EMI (Electro Magnetic Interference). With copper cables there is an extra cost to protect them from RFI and EMI. Fibers can carry data for approximately 200 miles without any repeaters. For copper cables, this limit is 30 miles [2]. Most important of all, the data carrying capabilities of an optical fiber are unparalleled. It has been shown that fibers can support data rates as high as 50 terabits per second [17].

Although data rates supported by optical fibers are very high, not all of it is well exploited right now. Current networks are not completely optical. These networks utilize a mixture of both electronic and optical technologies. Processing in the optical domain is not available yet. Hence, data received is converted to electronic form first, processed, converted back to optical form and then forwarded along optical fibers. These O-E-O conversions significantly limit the throughput of a network.

All optical networks represent a third generation of networks. Based on the technologies used, networks are classified into *first generation networks* - networks that use copper based technologies, *second generation networks* – networks using a combination of electronic and optical technologies and *third generation networks* where networks are completely optical where switching, buffering, processing and synchronization are all done in the optical domain.

Multicasting is a form of communication in which information is sent from a single source to multiple destinations. In wavelength division multiplexed (WDM) networks this is accomplished by the use of light trees and light forests. Optical fibers, the connecting media in optical networks, are prone to failures. Hence it becomes important to make the network resilient to link failures. This thesis presents two algorithms to overcome component failures in an optical network. The first algorithm considers cases of simultaneous multiple link failures while the second algorithm considers the situation of a node and a link fail at the same time.

## 1.1 Fiber Optics

An optical fiber is nothing but a dielectric optical waveguide made of glass. An optical fiber consists of three parts: the core, the cladding and the buffer. The core carries the light, the cladding confines the light to the core causing total internal reflection and the buffer forms an insulating wrap protecting the core and the cladding. Figure 1.1(a) shows the structure of a typical optical fiber. The core and the cladding have different refractive indices (RI). The RI of the core (n1) is always less than the RI of the cladding (n2), i.e., n1 < n2. Light propagates through the core because of total internal reflection (TIR). TIR occurs whenever the incident angle of the light is less than the critical angle [9]. The critical angle is defined as the angle of incidence at which light no longer gets refracted but instead gets totally reflected. This happens when light moves from a dense material to a less dense material. The concept of TIR is shown in Figure 1.1(b).

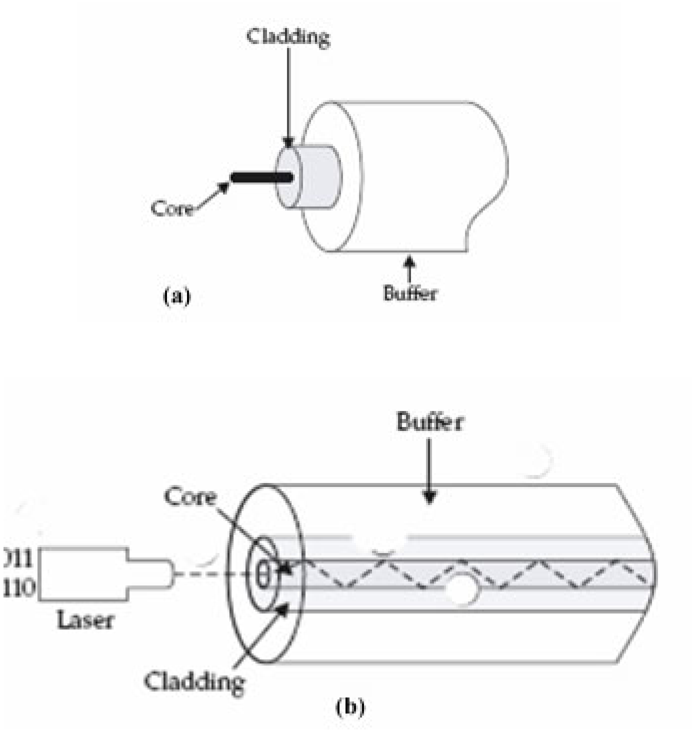


**Figure 1.1 An optical fiber (a) components (b) total internal reflection.**

A basic optical transmission system consists of three parts: the optical transmitter, the fiber and the optical receiver [2]. At the source, a serial bit stream is sent from its source to a modulator. The modulator consists of an light emitting diode (LED) or a laser that can be modulated according to the binary input stream to produce on/off light pulses.

These light pulses are injected into the optical fiber. The light then travels through the fiber, and at the receiving end, a detector coverts the light signal back into an electrical signal. The demodulator now decodes the electrical signal and the original bit stream is generated. As the optical signal travels along the fiber it may get attenuated due to a number of reasons. Hence repeaters might be required at certain intervals to amplify the signal. Figure 1.2 shows a basic optical transmission system.

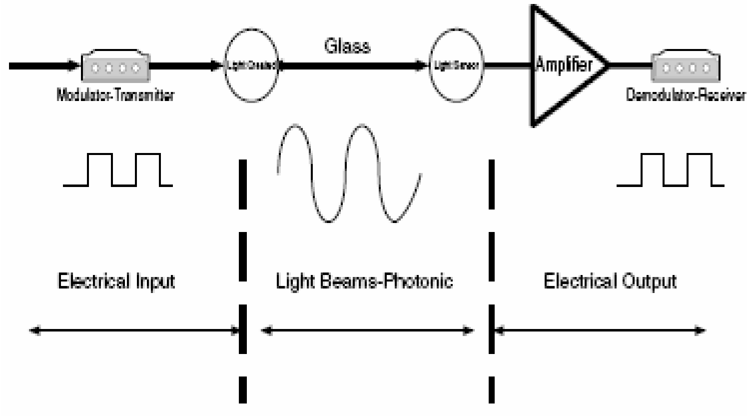
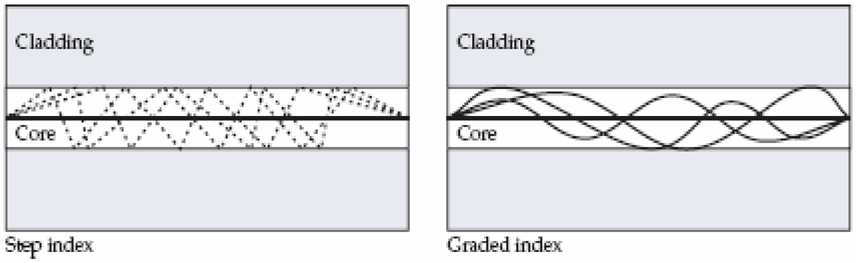


**Figure 1.2 A basic optical transmission system.**

Depending on the different ways in which a wave can propagate through a fiber, fibers fall into two categories: single mode fibers and multiple mode fibers. The core size of a single mode fiber is about 6-7 times the wavelength of the fiber [9]. This allows light to travel across a single path (mode) along the fiber. In this case the optical signal travels very long distances without much attenuation. Also a single mode fiber allows for higher transmission capacity. However, because of the small core diameter, coupling of the light into the core becomes somewhat difficult. Multiple mode fibers allow light to travel across different paths in the core of the fiber. Hence the light beams enter and leave the fiber at different angles. Multimode fibers are further classified as step index and graded index fibers depending on the index of the core and the way light travels within the core. The core of a step index fiber is made up of the same type of glass. Each mode of light travels in a different path resulting in high dispersion. Hence, the bandwidth of a step index fiber is limited. The graded index fiber has a core that is made up of different layers of glass. Light travels in parabolic paths along the core. The bandwidth capacity of a graded index fiber is about 100 times that of a step index fiber [9]. Figure 1.3 shows the cross sections of a step index and a graded index fiber.

**(a)**



**(b)**

**Figure 1.3 Different types of fibers: (a) multimode and single mode fibers, (b) step index multimode and graded index multimode fibers**

## 1.2 Wavelength Division Multiplexing (WDM)

Wavelength division multiplexing originated due to a need to exploit a larger portion of the available fiber bandwidth. Since it is difficult to utilize all the fiber bandwidth using a single high-capacity wavelength channel, WDM allows transmitting different light beams of different wavelengths along the core of the fiber. Prisms and diffraction gratings are normally used to multiplex and demultiplex different wavelengths.

Figure 1.4 shows a typical WDM system. Eight optical signals at different wavelengths from eight different transmitters are multiplexed together onto a single fiber. The add-drop multiplexer in the middle separates a particular wavelength, say λ4, which is destined for the local receiver, from the signal in the fiber and picks up another signal from the local transmitter at the same wavelength λ4. This signal is then merged with the other wavelengths passing through.

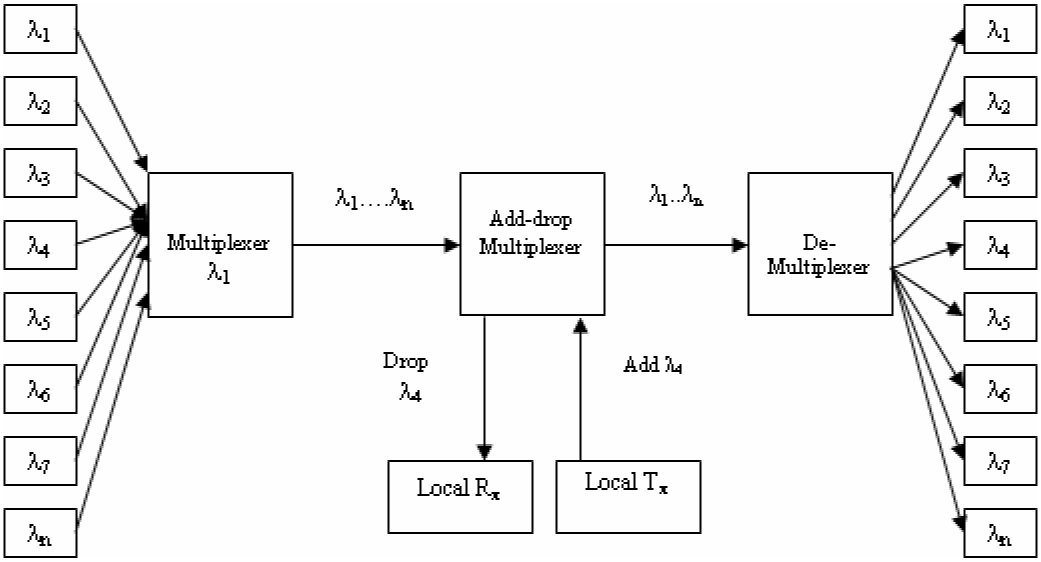


Figure 1.4 A simple WDM system.

The signal which now consists of eight different wavelengths then travels through the fiber. Upon reaching the demultiplexer at the destination, it is split into its constituent wavelengths and each wavelength is routed to its corresponding receiver [30].

## 1.3 A WDM System and its Components

Three major components of a WDM system are briefly described below.

### 1.3.1 An Optical Amplifier

An optical signal gets attenuated as it travels along the fiber due to a number of factors. Some of these factors are impurities in fiber glass, Rayleigh scattering, intermodal and chromatic dispersion etc. Also as the light pulses travel through the fiber they spread out thereby altering the duration of the pulses. In order to overcome these problems *repeaters* are used for every few hundred kilometers of fiber length. The repeaters *regenerate* the optical signals thereby reducing the degradation in quality of the signal as it travels through the fiber. There are three types of signal regenerators available. The 3R regenerator performs regeneration (simple amplification using erbium doped fiber amplifiers (EDFA)), reshaping (restores the original pulse shape of each bit) and reclocking (synchronizes the signal to its original bit rate and timing pattern). The 2R regenerator executes just regeneration and reshaping, while the 1R regenerator just takes care of amplification. WDM networks are a reality today due to the invention of EDFA amplifiers. Prior to EDFA amplifiers, electronic regenerators were used every few tens of kilometers. These regenerators converted the optical signal to electrical signals, amplified them and converted them back to optical signals. EDFA amplifiers amplify the signal optically regardless of individual bit rates, modulation schemes, or power levels. EDFA amplifiers are made up of several meters of silica glass fiber. This glass fiber is doped with a rare-earth element called erbium. The erbium ions are energized by an optical pump laser and these energized ions amplify (or boost) the optical signals traveling through the fiber. The amplified optical signal emerging from the EDFA retains all its original modulation characteristics [17]. An EDFA amplifier is shown in Figure 1.5.

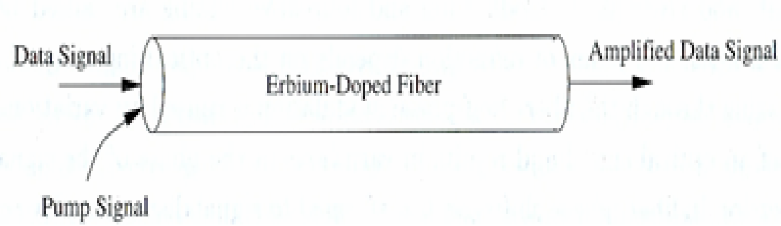


**Figure 1.5 An EDFA amplifier.**

### 1.3.2 Wavelength Add-Drop Multiplexer (WADM)

Sometimes it might be required to drop/add some traffic at certain intermediate points along the fiber. This is exactly what a WADM does. A typical WADM which is realized using a 2X2 switch is shown in Figure 1.6. It also contains a multiplexer and a demultiplexer. When the switch is in the *cross* state, the signal on the corresponding wavelength is dropped and a new signal is added on to the same wavelength. When the switch is in the *bar* state, then the signal passes through the WADM without being dropped [17]. Add/drop multiplexing can be done with optical or electronic signals. The device may deal only with wavelengths, or it may convert between wavelengths and electronic TDM signals. The setup in an add/drop multiplexer is generally static, and the device is not reprogrammed very often.

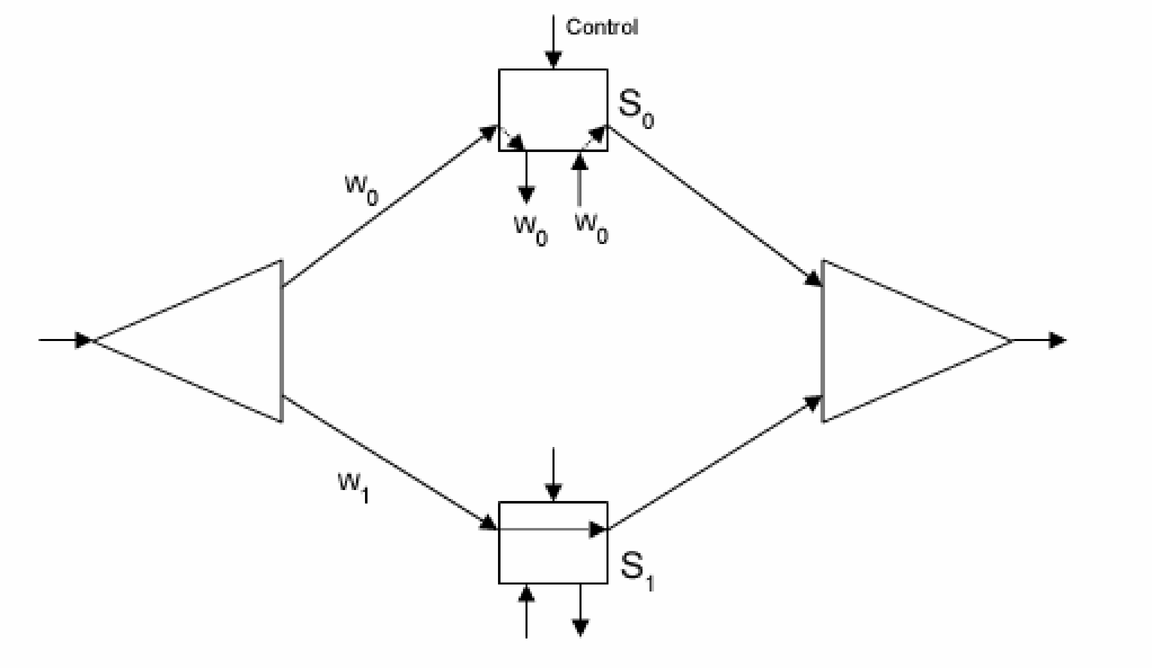


Figure 1.6 Wavelength add/drop multiplexer.

### 1.3.3 Wavelength Cross-Connects

Wavelength cross-connects are fiber interconnection devices. They accept different wavelengths on their input ports and switch (route) them to any of the output ports. The selection of the output port is usually made by the network control. Figure 1.7 shows a basic wavelength cross-connect. It is realized using optical switches, multiplexers and demultiplexers. A wavelength cross-connect performs switching at two levels: at the fiber level where all the wavelengths in a fiber are switched to an output fiber and at the wavelength level where only a few wavelengths in a fiber are switched from an input port to an output port. Such cross-connects are called *fiber switch cross-connects* and *wavelength selective cross-connects,* respectively. The most advanced of the cross-connects also perform wavelength conversions. When an incoming wavelength needs to be switched to an output port and the corresponding wavelength at the output port is not available then *wavelength interchange cross-connects* convert the incoming wavelength to one of the available wavelengths on the output port and performs the switching.

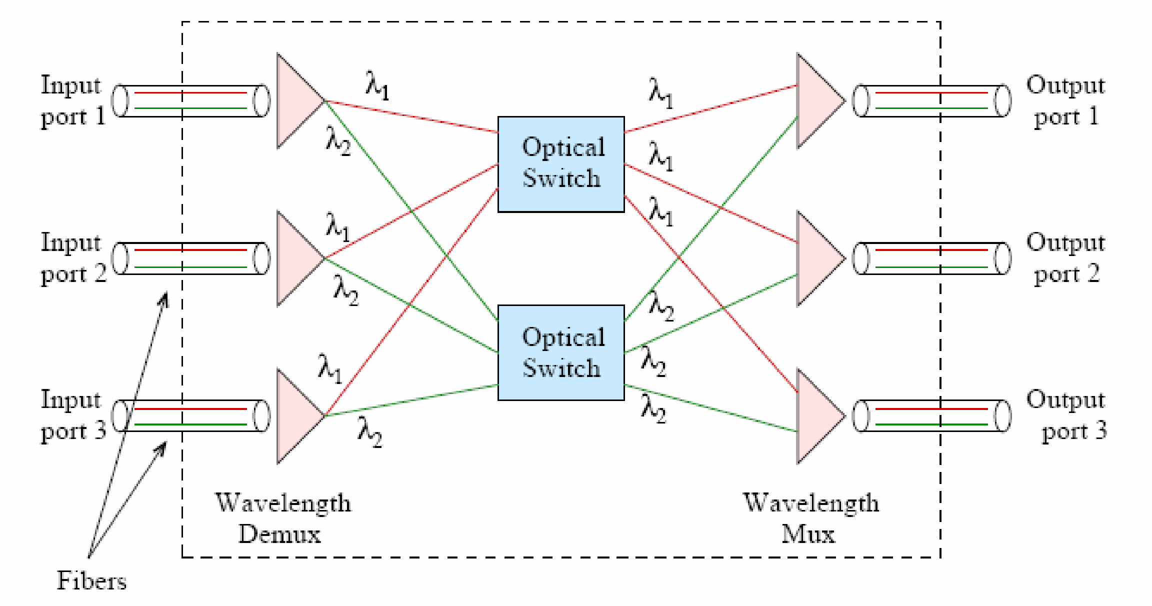


**Fig 1.7 A wavelength cross connect (2 λs per fiber) [21].**

## 1.4 WDM Networks and Their Classifications

WDM networks fall into three different categories. They are broadcast-and-select networks, linear lightwave networks and wavelength routing networks.

### 1.4.1 Broadcast-and-Select Networks

Broadcast-and-select networks are the simplest among WDM networks. At the heart of a broadcast and select network is a star coupler to which all the nodes in the network are connected. Each node consists of a set of tunable transmitters and tunable receivers. Whenever a node transmits at a particular wavelength, all the other nodes in the network tune their receivers to the transmitting node's wavelength in order to receive the message. In order to facilitate simultaneous transmissions, each node must transmit at a different wavelength. The star coupler combines all the incoming messages which are at different wavelengths and broadcasts them to all the other nodes. Since the receiving node would have its receiver tuned to a particular wavelength, it receives messages on that wavelength only. All other messages are discarded. Although such a network is simple to realize and has a *natural multicasting capability*, it requires a large number of wavelengths (as many as the number of nodes in the network) and cannot span long distances due to the loss of power associated with splitting [17]. Concurrent broadcasts have to be carefully planned to avoid collisions. Several protocols for scheduling simultaneous broadcasts and multicasts in such networks have been proposed in the literature [14][26].

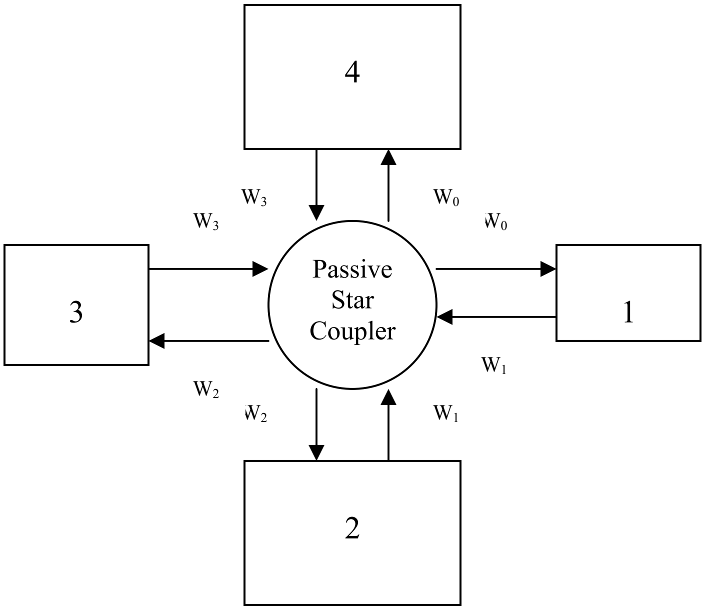


Figure 1.8 A simple broadcast-and-select network.

### 1.4.2 Linear Lightwave Networks

The optical spectrum can be partitioned into a number of wavelengths or groups of wavelengths called wavebands. Wavelength routed networks perform switching at wavelength level. However, switching can also be performed at the waveband level wherein groups of wavelengths are switched between the input and output port of a crossconnect. Such a concept is called waveband switching [6]. Figure 1.9 shows the partitioning of the optical spectrum into wavelengths and groups of wavelengths called wavebands. Wavebands and wavelengths within wavebands are separated by adequate guard bands for separation purposes [17]. With the advent of dense wavelength division multiplexing (DWDM), waveband switching has gained a lot of importance. Since switching is done at the waveband level, the hardware requirements of the optical cross-connects (OXC) (for example number of ports) significantly reduces thereby resulting in a simplified architecture for the OXCs. Multi-layer and single layer architectures have been proposed for OXCs to support waveband switching [4][5].

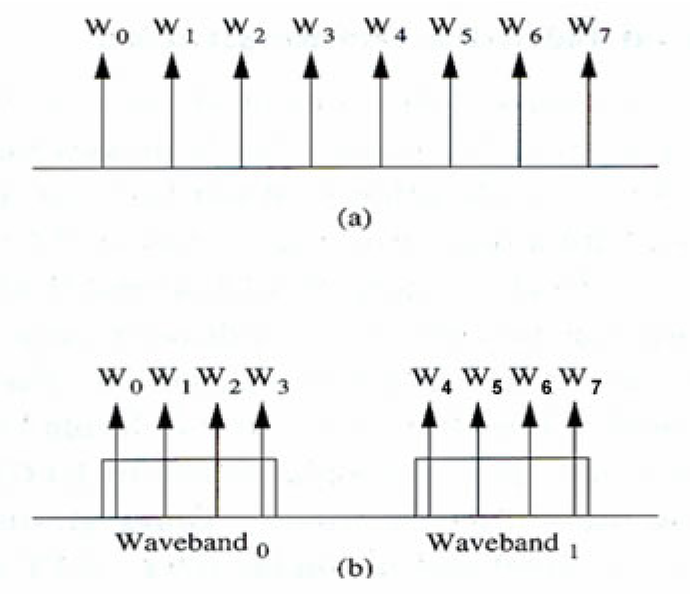


Fig 1.9 Wavelength and waveband partitioning.

Wavelength continuity constraints and distinct wavelength assignment constraints are applicable to linear lightwave networks. Two additional constraints that are usually applied are *inseparability* and *distinct source combining.* The inseparability constraint states that channels belonging to the same waveband when combined on the same fiber cannot be separated within the network [17]. The distinct source combining constraint states that only signals from distinct sources can be combined on the same fiber [17].

### 1.4.3 Wavelength Routed Networks

Wavelength routed networks overcome most of the problems associated with broadcast and select networks. Nodes in wavelength routed networks communicate with each other by establishing a wavelength continuous path called *a lightpath.* A lightpath is an all-optical communication path established between the source and the destination nodes by allocating the same wavelength throughout the route of the message [17]. Hence there are no optical-electronic-optical conversions or buffering in the intermediate nodes. This process is known as *wavelength routing*. The necessity of using the same wavelength along the entire path from the source to the destination is called the *wavelength continuity constraint.* Also two lightpaths cannot be assigned the same wavelength on any fiber. This is called *wavelength assignment constraint* [17]. A wavelength routed network with five nodes and two wavelengths per node is shown in Figure 1.10 [17]. Every fiber link in the network carries physical paths corresponding to two lightpaths. Paths $p_0$, $p_1$, $p_2$ and $p_3$ are lightpaths established between node pairs *<0,2>, <1,3>, <2,4>* and *<3,0>*, respectively.

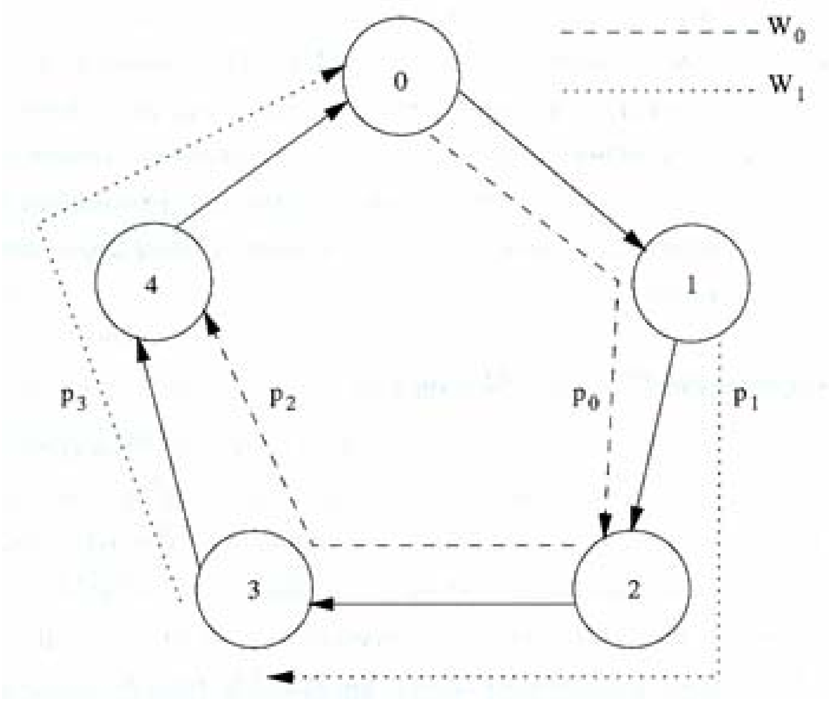


**Fig 1.10 A wavelength routed network.**

It can be clearly seen that lightpaths support *wavelength reuse*. Although the wavelength continuity constraint creates a high bandwidth pipe transferring data at Gbps speeds, it has its own drawbacks. Consider node *4* being the source and node *1* being the destination. Although $w_0$ is free on link *<4-0>* and $w_1$ is free on link *<0-1>,* a lightpath cannot be established because of the wavelength continuity constraint. This results in loss of bandwidth.

#### 1.4.3.1 The RWA Problem in Wavelength Routed Networks

Since lightpaths form the basic building blocks of a wavelength routed network, effective establishment of lightpaths is important. Provisioning of routes to lightpath requests and assigning wavelengths on each of the links along these routes from the possible choices so as to optimize a certain performance metric is called the *routing and wavelength assignment* (RWA) problem [19]. The wavelengths should be assigned in such a way that no two lightpaths sharing the same physical link should use the same wavelength on that physical link. Several RWA algorithms have been proposed in the literature that differ in their assumptions [8, 11, 20, 21]. Some works assume *static* traffic patterns and that the lightpaths are to be established in the network beforehand. Others assume a *dynamic* traffic pattern, where the requests to establish lightpaths arrive in random order in real time [19]. The bandwidth losses associated with the wavelength continuity constraint can be overcome by using wavelength converters [19]. Certain RWA algorithms proposed in the literature consider networks that have both sparse wavelength conversion (only a subset of the nodes are capable of wavelength conversion) and full wavelength conversion (all the nodes in the network are equipped with wavelength converters).

The RWA problem is dealt with by decomposing it into the routing problem and the wavelength assignment problem and approaching them separately.

Various approaches for routing connection requests are as follows [28].

1. **Fixed routing:** In fixed routing the same route is always used between the given source-destination pair. An example of this is *fixed shortest-path routing,* where the shortest-path between the pair of nodes is established offline and any connection between this node pair is established using this predetermined route.

2. **Fixed alternate routing:** This involves considering multiple routes between a given source-destination pair. Every node is required to maintain a *routing table* consisting of an ordered list of a number of fixed routes to each destination node.
3. **Adaptive routing:** The route from source to destination is chosen dynamically depending on network conditions.
4. **Fault tolerant routing:** Routes are established between a source-destination pair considering link and node failures. A common approach is to set up two link-disjoint paths, one primary that is used for transmission and the other secondary that is used as a backup-path.

Some of the wavelength assignment heuristics include the following [28].

1. **Random wavelength assignment:** From the pool of all wavelengths that are available on a given route, a wavelength is picked at random.
2. **First-Fit:** The available wavelengths are numbered. When the available wavelengths are being searched, a lower numbered wavelength is considered before a higher numbered wavelength, and the first available wavelength is then selected.
3. **Least-used/SPREAD:** The least-used wavelength in the entire network is chosen so as to balance the load among wavelengths.
4. **Most-used/PACK:** Selects the most-used wavelength in the network.
5. **Round-robin:** Wavelengths are assigned in a round-robin manner from the set of available wavelengths.

The above heuristics are defined for a single fiber network. Separate heuristics like *Min-Product*, *Least-Loaded* and *MAX-SUM* exist for multi-fiber networks.

## 1.5 Future of WDM Networks

The ultimate goal of advancements in optical networks is to have end-to-end wavelength services [12]. Three major advancements to watch out for in the future are: *all-optical packet switching*, *dense wavelength division multiplexing* and *wireless optical networks (WON)*. An all-new optical layer in the telecommunications network promises a revolution. Optical-layer technology will increase network capacity, allowing network providers to transport more than 40 times the traffic on the same fiber infrastructure [12]. This would lead to lower costs and ensure affordable bandwidth. The increased capacity provided by the optical layer will cause expensive high-bandwidth services such as videoconferencing to the desktop (or home), electronic commerce, and high-speed video imaging to become affordable. Figure 1.11 projects an outlook for optical network based services [18].

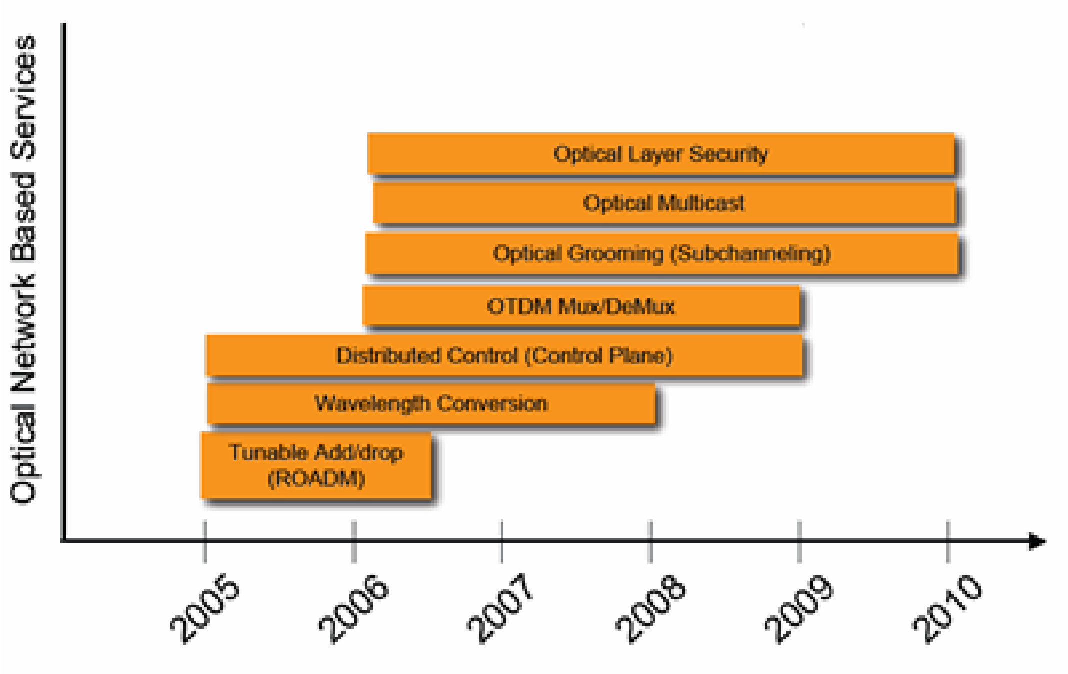


**Fig 1.11 Future of optical networks.**

# Chapter 2

# Multicasting in Optical Networks

There are basically three different types of communication models. They are *unicast*, where communication is between one source and one destination, *broadcast,* where communication is between one source and all other nodes and *multicast,* where communication is between one source and multiple but not all other nodes. Some applications of multicast communications are multimedia, teleconferencing, databases, distributed computation, real-time workgroups etc. As we have seen in Chapter 1, a simple broadcast-and-select network, i.e., a one-hop network, can be used to implement multicasting. However, due to power budget constraints, multicasting is better suited for multi-hop networks.

Multicasting in optical networks can be accomplished in three different ways. They are:

1. Multiple copies of the message – Make multiple copies of the same message electronically and transmit them to the destinations. Although this is a very simple method of accomplishing multicasting, it is not efficient because of the numerous optical-electronic-optical conversions necessary at every node.
2. Multiple unicasts – In this approach multicasting is achieved by conducting unicasts between the source and each destination, i.e., messages are routed from the source to each destination separately. This method is also not efficient since it consumes a lot of bandwidth. The bandwidth consumption is mainly dependent on

the number of destinations in the multicast and grows as the number of destinations increases.

3. Light splitting – This approach involves making multiple copies of the message optically. This approach is the most efficient of the three approaches.

In multicasting terms, a *tree* refers to a set of destinations connected together with the source of the multicast as the root. A source transmits a message only once to communicate to all the destinations that are the part of the same tree. Multicasting in optical networks can only be supported when nodes in the network are capable of splitting light. Sometimes not all nodes in the network are capable of splitting light. Such networks are called *sparse split networks*. In sparse split networks it may not be possible to include all the destinations in one tree. Hence, a multicast session then may be made up of a number of trees. All such trees put together are called a *multicast-forest.* When all the nodes in the network have splitting capability, the network is called a *fully split network*.

## 2.1 Multicast Routing

Figure 2.1 shows the concept of multicast routing in optical networks. Node *s* is the source of the multicast and nodes *d1, d2, d3* and *d4* are destinations. Figure 2.1(a) shows multicast routing in absence of light splitting capability at node *d2* while Figure 2.1(b) shows routing in the presence of light splitting capability at node *d2*. It can clearly be seen that in absence of optical splitting there are three multicast *trees (s→ d1), (s→d2,* *d2 → d3)* and *(s → d2 → d4)* as opposed to two multicast trees *(s→ d1)* and *(s→d2, d2 → d3 and d2 → d4)* in Figure 2.1b.

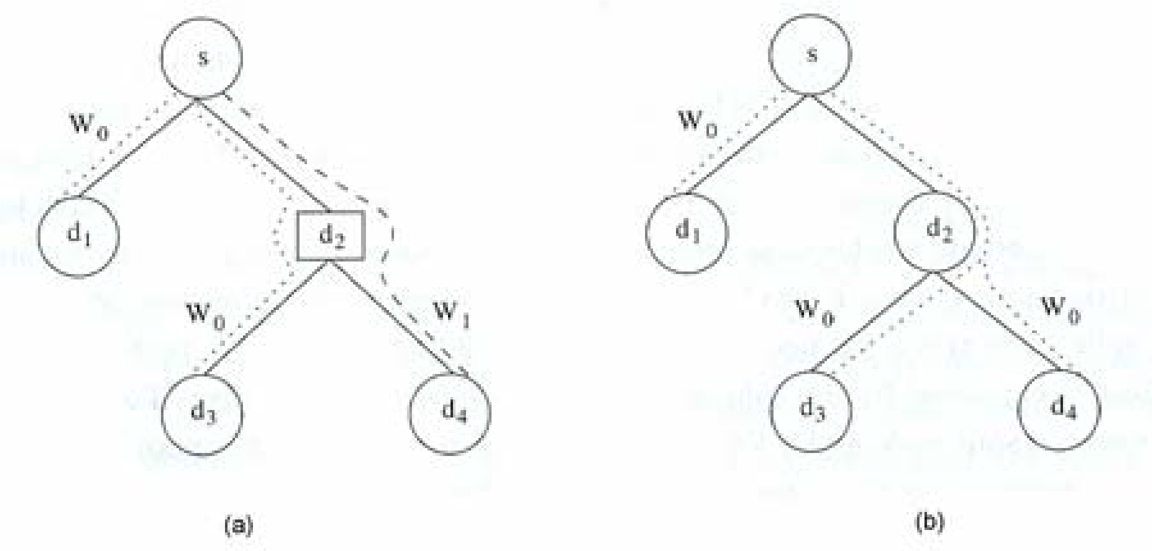


**Fig 2.1 Illustration of multicasting with (a) no splitting and (b) splitting.**

### 2.1.1 Light Trees and Light Forests

A *light tree* [22] is similar to a multicast tree except that a single wavelength is used for all the links in the tree. From Figure 2.1(b) we can see that the same wavelength, $W_0$, is used along the entire tree to deliver data to all destinations. Sometimes it might be required to use more than one light tree to service all destinations. A *light forest* [29] is a collection of light trees. In Figure 2.1(a), $W_0$ in used along *(s→ d1)* and *(s→d2, d2 → d3)* and $W_1$ along *(s → d4)*.

## 2.2 Node Architectures

Nodes in optical networks that support multicasting are broadly classified into two categories depending on their splitting capabilities. A node that can split light is called a *multicast-capable* node, or an MC node. Those that cannot split light are called *multicast-incapable* nodes, or MI nodes. Figure 2.2 shows the architecture of an MC node.

### 2.2.1 MC Node Architecture

An MC node is made up of optical power splitters. The architecture of one such node is shown in Figure 2.2 [17]. It has two input links, two output links and three wavelengths per fiber. Hence the incoming signal is split onto 2X3 output connections. In stage 1, a signal from each input link is split into two different signals. This ensures a particular output link is selected from the set of output links connected to the node. Stage 2 employs a 1:3 splitter. This splitting is to choose the wavelengths onto which the input signal needs to be transmitted. The output of the 1:3 splitters is then fed to a set of space division switches (SD switches). The SD switch selects one of the two input signals and forwards it to a tunable filter (TF) to extract the signal on a particular wavelength. The wavelength converters (optional) avoid wavelength clashes at the output. For every output link there is a multiplexer so that different wavelengths are multiplexed and transmitted onto the same fiber. Semiconductor optical amplifiers (SOAs) are used in the end to compensate for the power lost due to splitting. This cross connect employed by an MC node is called splitter-and-delivery cross-connect. (SaD). An advanced version of the SaD crossconnect is proposed in [1]. It is called the multicast-only splitter-and-delivery (MOSaD) cross connect. The MOSaD architecture uses power splitters for multicast connections only, allowing unicast connections to pass without enduring unnecessary power losses.

### 2.2.2 MI Node Architecture

A node without splitting capability is called a multicast incapable node. MI nodes tap a small fraction of the signal and switch the remaining signal as it is to the neighbor.

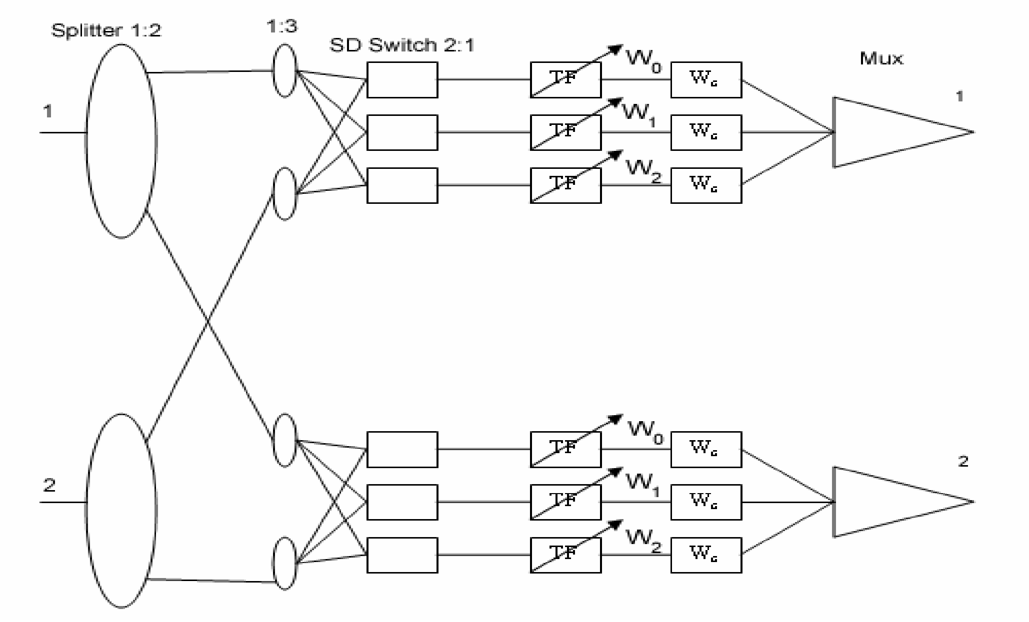


Figure 2.2 Architecture of an MC node.

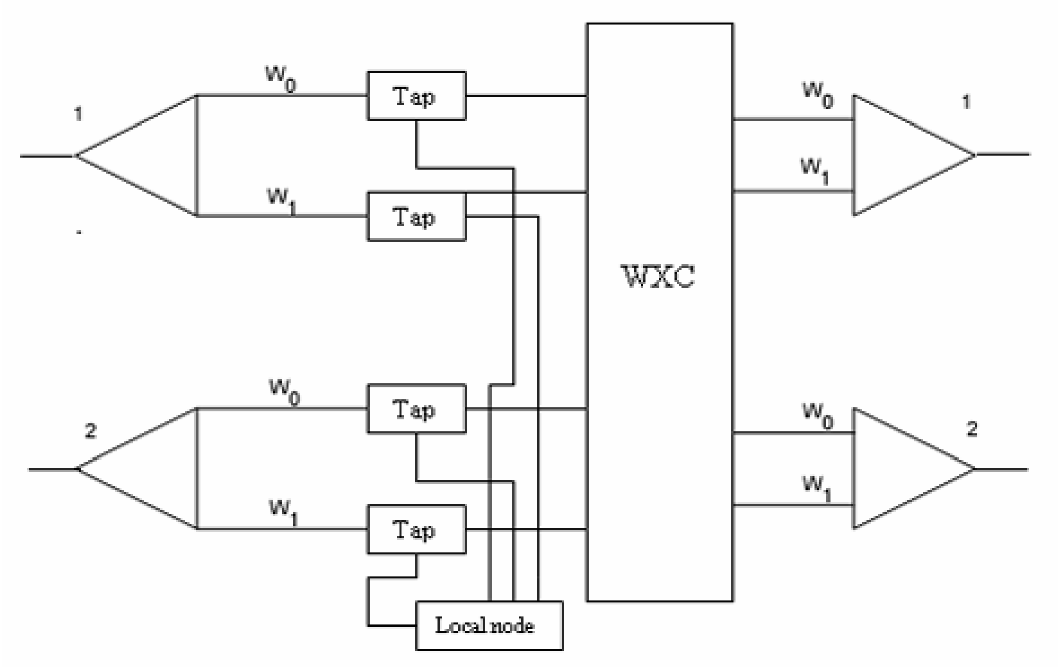


Figure 2.3 A drop-and-continue node.

Hence, it is called a drop-and-continue (DaC) node. The tapped signal is used by the local station. The architecture of a DaC node is shown in Figure 2.3. The input signal is demultiplexed, and the demultiplexed signals are fed to a drop module that taps around five percent of the signal. The output of these drop modules is then switched by an OXC found in any other wavelength routed node.

## 2.3 Multicast Tree Generation

In optical networks, multicast route determination is formulated as a problem of lighttree construction. An objective of the tree design is to minimize data replication by allowing packets to be optically replicated only at branches of the tree [17]. In a fully split network a single tree can be generated to include all destinations. Hence tree formation here is similar to that of conventional networks. Optimizing tree formations may be subject to certain constraints like minimizing the average packet hop distance, minimizing the total number of transceivers in a network etc [17]. In the case of sparse-split networks, the objective of tree generation algorithms might be to minimize the number of wavelength channels, minimize the number of wavelengths in a fiber, minimize the delay etc. Two approaches to generate multicast tree/forest in sparsely split networks are as follows.

a. **Source rooted approach** - where the source of the multicast is the root of the tree. The objective here is to reduce the cost (number of wavelength channels or hops) of the path between the source and the destination. Based on the objective, this method is further sub-divided.

1. **Source-based tree generation** - The multicast tree is constructed in such a way that the cost of the individual paths from the source to destination is minimized. The *re-route-to source* [17], *re-route-to-any* [17] and *member-first* [17] algorithms fall under this category.
2. **Steiner based tree generation** - The tree is constructed so as to minimize the overall cost of the tree. This problem is known as minimum Steiner tree problem and has been shown to be NP-complete [13]. The *member-only* [15] algorithm adopts this approach.

**b. Virtual-source rooted approach** – Here the tree is constructed with a special node called the virtual source at the root of the tree [25]. These algorithms perform better than algorithms employing a source-rooted approach because of better wavelength channel utilization.

## 2.4 Failure Recovery in Wavelength Routed Networks

A wavelength routed network, just like any other network, is prone to component failures. Since an optical network carries huge volumes of traffic, failure of even a single component can result in severe data loss. *Fault tolerance* is defined as the network's ability to reconfigure itself and reestablish communication upon a component failure [17]. *Restoration* is defined as the process of rerouting affected traffic after detecting a failure. A network that is capable of handling failures is called a *survivable* or a *restorable* network. Designing such survivable networks requires use of spare resources. Various schemes have been proposed in the literature to combat failures. These restoration schemes are characterized by the functionality of wavelength cross-connects, resource sharing, means of network control and failure model.

A classification of the different restoration methods is shown in Figure 2.4. We consider wavelength-routed networks only and hence the restoration methods discussed from here on talk about restoring a lightpath. Lightpath restoration methods can be broadly classified into two categories: *proactive* methods and *reactive* methods. Reactive methods are probably the simplest methods for recovery. Upon failure of a lightpath, a search to find a new lightpath that doesn't incorporate the failed components is initiated. Such a method, although simple, has its advantages and disadvantages. The advantage is that it has a low overhead in absence of failures. However, a successful recovery cannot be guaranteed all the time because of the dynamic nature of the restoration method. At the time of establishing a new lightpath, there might not be sufficient resources available thereby resulting in this type of method failing. Also in case of simultaneous recovery attempts for different failed lightpaths, several retries may be required thereby increasing network traffic and restoration time.

Proactive methods overcome the shortcomings of reactive methods. Here backup lightpaths are identified and reserved offline. Hence there is a hundred percent *restoration guarantee.* Restoration guarantee refers to the guarantee with which a failed path finds its backup path readily available upon a failure [17]. The backup path then acts as a substitute for the primary lightpath by allowing the traffic to be re-routed through it rather than the failed primary lightpath.

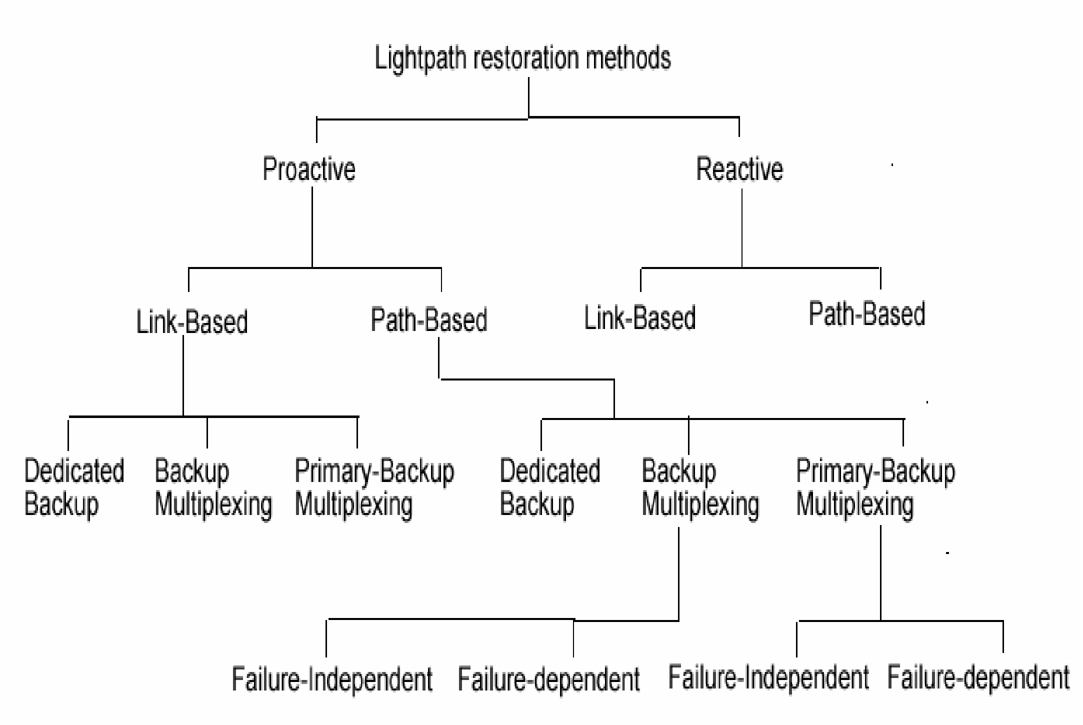


**Figure 2.4 Classification of restoration methods.**

Since the backup paths are already established even before the failure actually occurs, proactive methods have a shorter restoration time compared to reactive methods.

Proactive or reactive restoration methods are further classified into *link-based restoration* methods and *path-based restoration* methods. Link-based methods use *local detouring* while path-based methods use *end-to-end detouring* [17]. In link-based restoration, traffic is rerouted between the end nodes of the failed link. This path between the end nodes of the failed link together with the working segment of the primary path is used as the backup path. In link-based restoration, the number of possible backup paths is limited and backup paths are usually long. Also, handling node failure by this method is very difficult. In path-based restoration, a backup lightpath is established between the end nodes of the failed primary lightpath. This method has the following advantages. It can use a wavelength other than the one used in the primary lightpath along the backup path. Also it allows for better resource utilization. Figures 2.5 and 2.6 illustrate link-based and path-based restoration methods. $p_1$ is the primary lightpath while $b_{11}$ and $b_{12}$ are the back-up paths. Upon failure of the link *0* → *1* traffic is routed around nodes *0* and *1* through backup path *0* → *3* → *4* → *1* i.e. $b_{11}$ and when link *1* → *5* fails traffic is routed around nodes *1* and *5* through backup path *1* → *2* → *5* i.e $b_{12}$.

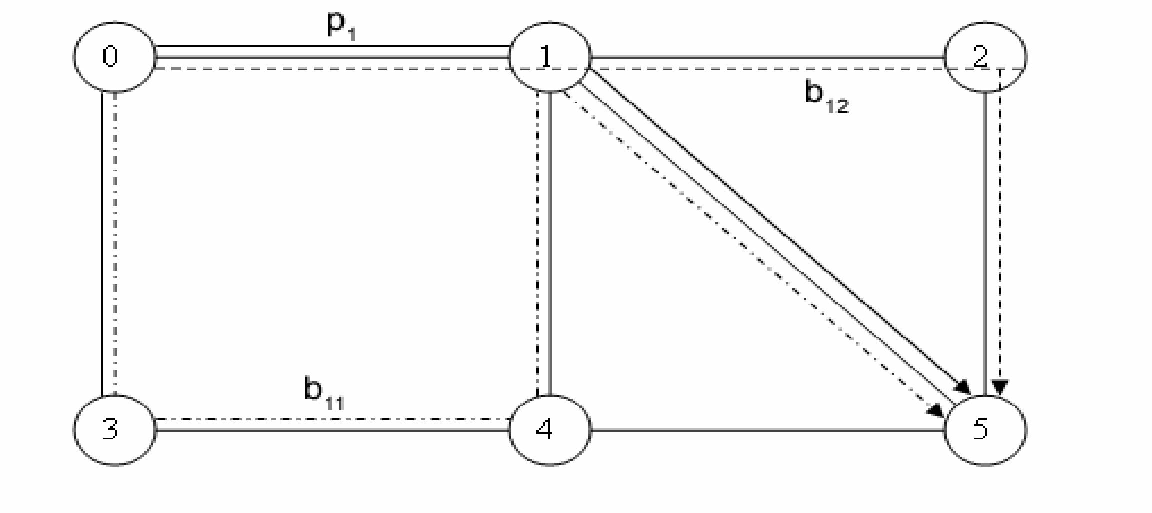


**Fig 2.5 Link based restoration.**

In path based restoration shown in Figure 2.6 the backup path $b_1$ i.e *0* → *3* → *4* → *5* is established between the end nodes *0* and *5* of the failed primary lightpath $p_1$ i.e. *0* → *1* → *5.*

Dedicated backup lightpaths, which do not share wavelength channels between them, can be used to as a proactive method to recover from failure. This method, although fast, ends up reserving a lot of resources. Figure 2.7 shows dedicated backup paths $b_1$ and $b_2$ for lightpaths $p_1$ and $p_2$ respectively.

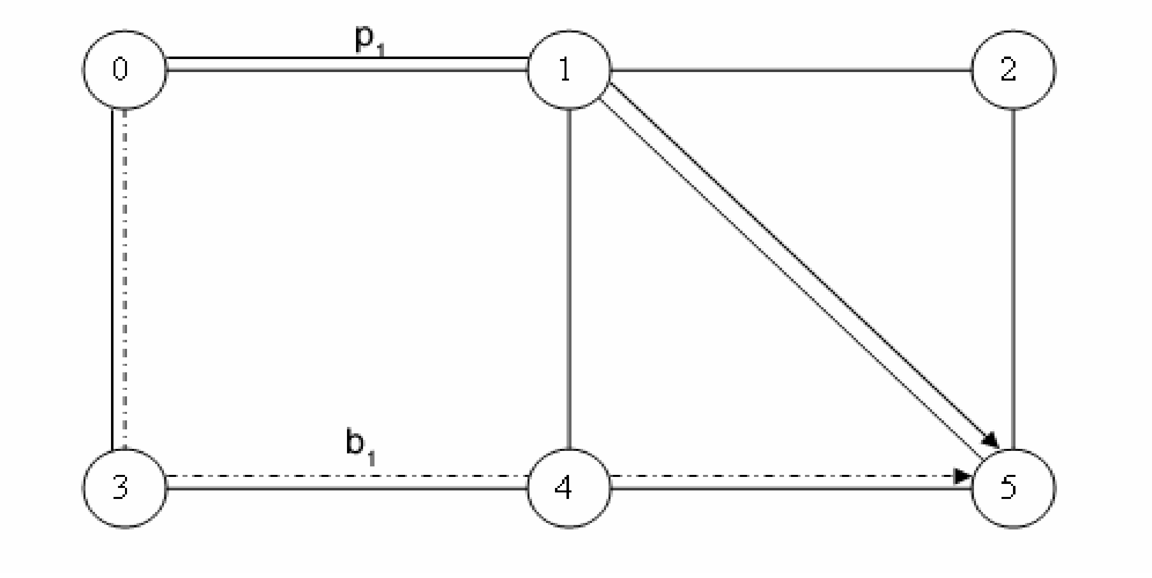


**Fig 2.6 Path based restoration.**

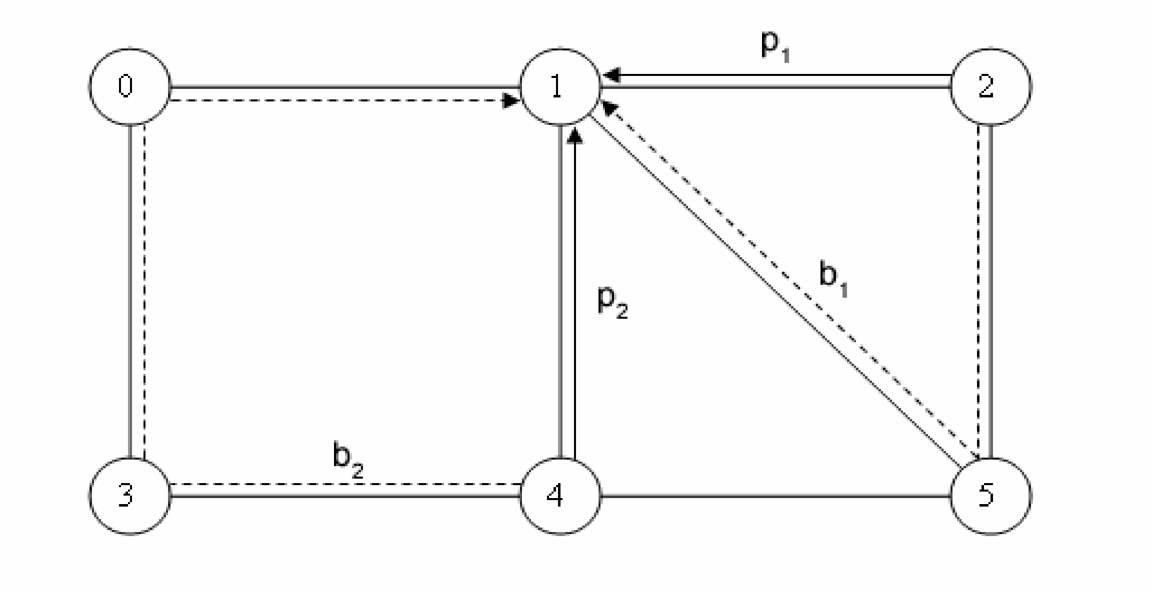


**Figure 2.7 Dedicated backup path restoration.**

Path-based restoration methods are either *failure-dependent* or *failure-independent* [17]. A failure-dependent method has a backup lightpath associated with the failure of every link used in the primary lightpath. In this case the backup lightpath can utilize links present in the primary lightpath, except for the failed link. In the case of failure-independent methods, backup lightpaths are link disjoint with the primary lightpaths. Hence, the backup path is used irrespective of which link fails in the primary lightpath. This results in poor resource utilization.

*Backup multiplexing* and *primary-backup multiplexing* allow wavelength channels to be shared between backup paths. The backup multiplexing technique is shown in Figure 2.8. Paths $p_1$ and $p_2$ are link-disjoint primary lightpaths. If they do not fail simultaneously, then $b_1$ and $b_2$ can share the same wavelength on link *5 → 1*. Link 5 → 1 is used by $b_1$ when $p_1$ fails and by $b_2$ when $p_2$ fails.

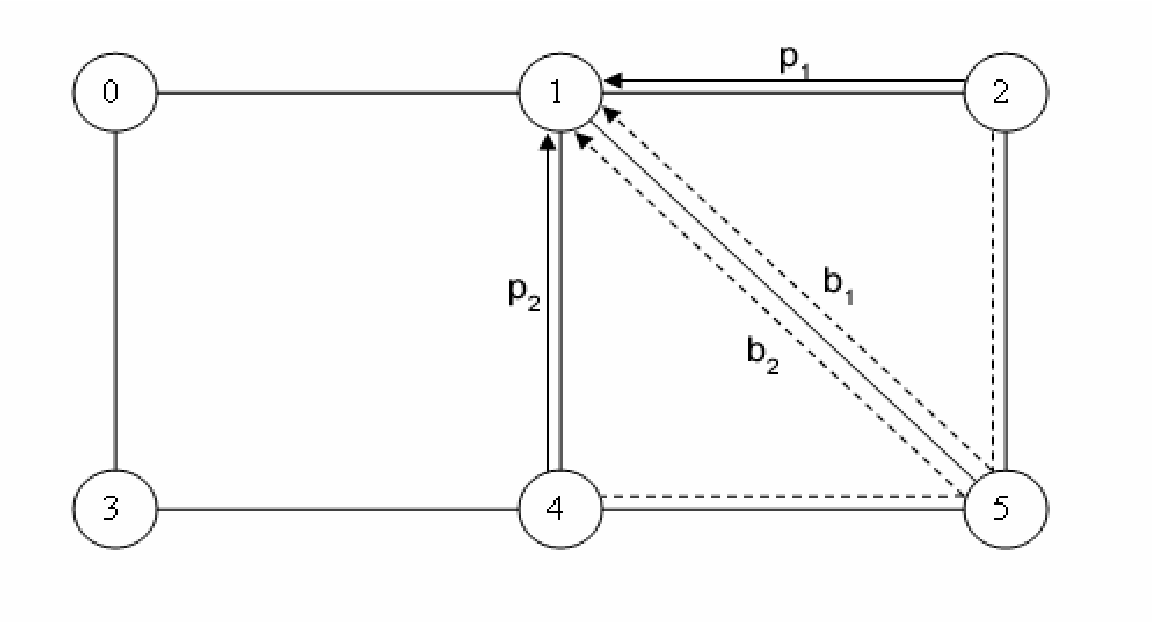


**Figure 2.8 Backup multiplexing restoration.**

Primary-backup multiplexing allows a primary lightpath and one or more backup lightpaths to share a channel [16]. This technique is suited more for dynamic traffic patterns where the average lifespan of a lightpath is very short. It allows larger number of lightpaths to be established at the cost of restoration guarantee. This technique is shown in Figure 2.9. The figure shows three primary lightpaths are $p_1$, $p_2$ and $p_3$ and their backup paths are $b_1$, $b_2$ and $b_3$ respectively. Since primary lightpaths $p_1$ and $p_2$ are link-disjoint their backup paths $b_1$ and $b_2$ share the channel on link 4 → 1. Link 1 → 2 is shared by $p_3$ and $b_1$ and link 5 → 2 is shared by $p_1$ and $b_3$. Due to this kind of sharing $p_1$ and $p_3$ are not recoverable when they fail simultaneously. However, as said earlier, since the lightpaths are short-lived, once one is terminated the other one becomes recoverable. Primary-backup multiplexing assumes that, practically, faults are not frequent enough and that every lightpath need not have fault tolerance capability to ensure network survivability [17]. Hence the restoration guarantee for such a scheme is less than one hundred percent.

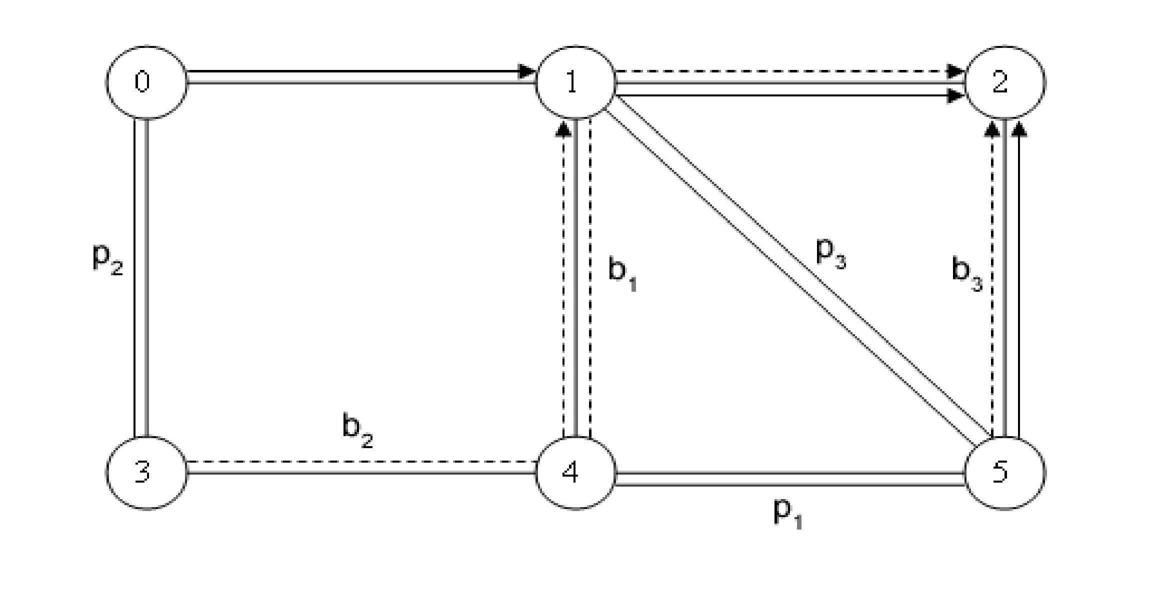


**Figure 2.9 Primary-backup multiplexing.**

## 2.5 Backup Path Computations in Multicasting

Backup path computation in case of multicasting can be slightly different from the other cases. Here we describe some of the strategies for computing backup paths for multicasting as listed in [23]. If both the primary and the backup trees are totally edge disjoint, then failure of an edge cannot affect both trees. However, it might not always be possible to find a backup tree that is edge-disjoint with the primary tree. Consider the network shown in Figure 2.10. Node 1 is the source of the multicast while nodes 2, 3, 4, 5 and 6 are destinations. The primary paths are indicated in bold lines. Here it is impossible to find a backup tree that is edge disjoint with the primary tree. However if we relax this condition to finding an arc disjoint tree, then a backup tree can indeed be found by employing the property of *directed link disjointness*. This requires that the primary and backup trees not share any link in the same direction. Since all the links are bidirectional this property holds good.

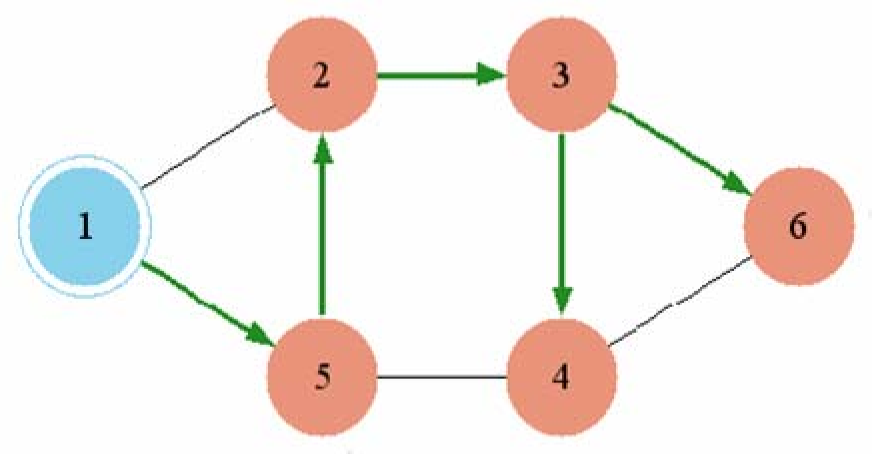


**Figure 2.10 A multicast session.**

If we are unable to find a backup tree that was totally edge disjoint with the primary tree, then removing all the primary edges from Figure 2.10 would lead to an unconnected graph depicted in Figure 2.11. Figure 2.12 shows the backup tree derived from the property of directed link disjointness. The backup path is shown in dotted lines. It can be seen that the primary and the backup trees do not share the same direction on any link.

The rest of this thesis is organized as follows. Chapter 3 describes problem formulation, the fault model, the assumptions and the algorithms to overcome multiple link and node failures in optical networks. The way the algorithms work is illustrated

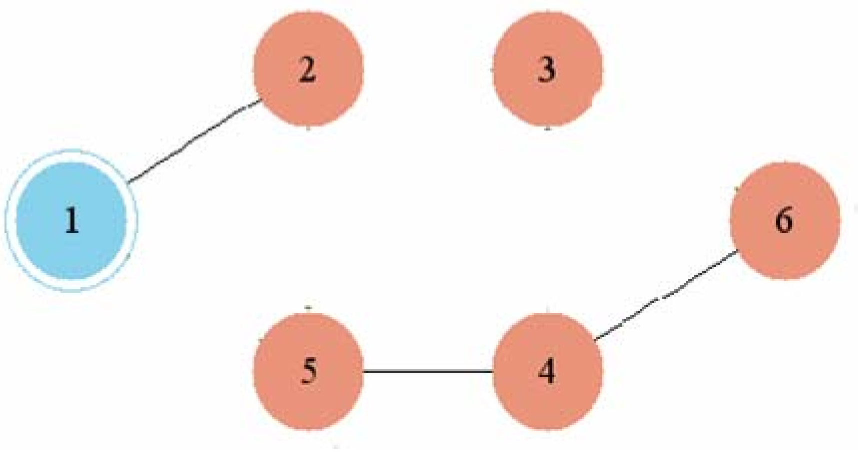


**Figure 2.11 Disconnected graph with primary path removed.**

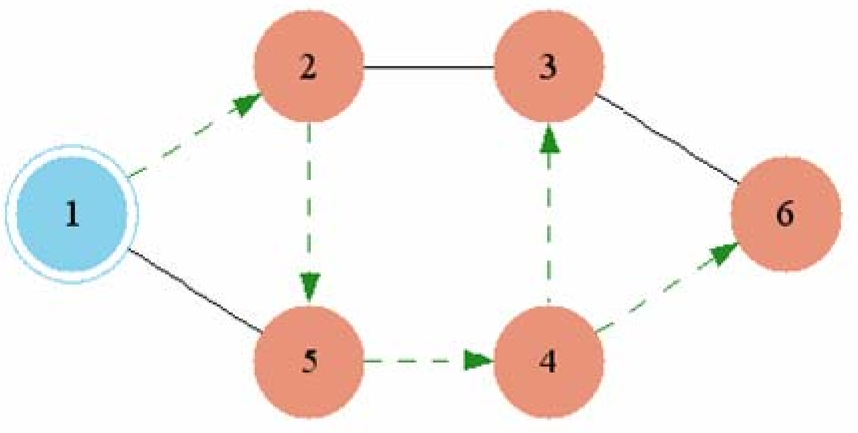


**Figure 2.12 Directed link disjointness.**

with examples. Chapter 4 presents the results obtained through simulations. Chapter 5 gives the conclusion.

# Chapter 3

# Survivable Multicasting

Optical fibers, the connecting media in optical networks, are prone to failures. A single fiber cut can result in loss of huge volumes of data. A fiber cut, possibly the result of an errant excavation, has been estimated to occur, on average, once every four days by TEN, a pan-European carrier network [10]. Hence it becomes important to make the network resilient to link failures. *Survivable multicasting* refers to making a multicast session resilient to component failures such as link and node failures. Numerous algorithms based on heuristics exist in the literature to combat single link failures in generic networks. Events like major catastrophes, terrorist attacks, natural disasters.etc, can trigger multiple link failures. This thesis deals with making a multicast session survivable, possibly, under conditions of multiple link failures and node failures.

## 3.1 Problem Definition

As mentioned in Chapter 2, multicast sessions are based on light trees that are established depending on the light splitting capabilities of the nodes in the multicast session. Link failures can stop the data from being received by several destination nodes in a multicast session. Hence, making the multicast session resilient to link failures is very important. In this thesis we present new algorithms that make a multicast session survivable to simultaneous multiple link failures. We also consider the case of having a node and a link fail at the same time. The algorithms presented in this thesis can not only handle single link failures but can also combat simultaneous multiple link and node failures.

The problem of making multicast sessions survivable to single link failures has been studied in [23] and [24]. It was shown in [24] that establishing primary and backup trees at minimal cost in such a way that both trees do not share a link in the same direction is an NP-complete problem. [24] considers sparse-split networks and uses ILP formulations. ILP formulations are suitable for small-to-moderate sized networks. For large networks we need to have efficient algorithms that generate quality solutions in reasonable time [23].

The closest work addressing a problem similar to the one investigated here is reported in reference [23]. The work in [23] shows that both proactive and reactive schemes can coexist in survivable networks. It defines a parameter called *"threshold"* that controls the amount of proactive and reactive recovery in multicast sessions. The work also explores the possibility of resource sharing among backup paths. However, the work in reference [23] does not consider the possibility of multiple link failures or node failures. It assumes only a single link failure at any point of time, and the algorithms are designed to deal with single link failures. The network considered in [23] is a sparse splitting network.

In this thesis, we assume that the all nodes in the network have splitting capability. For the first time, we use the novel concept of *minimal-hop cycles* to combat multiple link failures in a multicast session in optical networks. We also study the efficiency of our schemes in terms of power delivery.

The solutions proposed in this thesis revolve around the concept of using minimal hop cycles to establish backup paths that utilize the bidirectional property of the fiber links. Our solution is a hybrid scheme in the sense that it is a combination of proactive and reactive schemes. Backup paths are calculated and stored offline. When a link failure is detected, the backup path is retrieved and routing tables are updated accordingly. This hybrid scheme also combines the link-based and *failure-dependent* path-based restoration concepts.

### 3.1.1 Resource Sharing among Backup Paths

Resource sharing was explored in reference [23] for reuse of network resources. The algorithms presented here consider one link in the multicast session at a time to find a cycle to construct a backup path. For each such link a cycle is found which can provide a backup path in case that link fails. During the discovery of these cycles we allow resource sharing. If another link in the multicast session happens to fall within an already chosen cycle, then the two multicast links are allowed to share the same cycle. In other words, all the links that are a part of the multicast session that fall in the same cycle will have backup paths within that same cycle. The advantages of such a resource sharing technique are:

1. Reusing the same network elements to provide protection for more than one link in the primary multicast session.

2. The backup paths for all the multicast session links that fall within the cycle need not be computed. Hence, backup path computation time is reduced.

Formation of minimal hop (shortest path) cycles is explained in detail in Section 3.4.

## 3.2 Assumptions

Our work is based on the following assumptions

1. All the nodes in the network have splitting capabilities,
2. The network is at least 2-connected so that failure of a link doesn't disconnect the network,
3. All the links in the network are bidirectional and
4. The number of wavelengths available for backup paths on a fiber is greater than the number of multicast sessions so that different multicast sessions can be protected simultaneously.

## 3.3 Fault Model

Unless otherwise specified, the following fault model is assumed

Failure type – Single and simultaneous multiple link failures

A failed link is equivalent to a non-existent link in the network.

Failure mode – Dynamic

Failures are known as and when they occur.

Failure Neighborhood

A link failure affects the rest of the multicast session downstream.

Failure Recovery

A hybrid scheme that is a combination of proactive and reactive schemes is used to recover from failures. A backup path for the failure exists within the cycle associated with the link that has failed. Upon failure of a link, the stored cycles are probed to identify the cycle to which the link belongs. Then backup paths are extracted from the cycle and routing information is updated accordingly.

## 3.4 Minimal-Hop (Shortest-Path) Cycles

The formation of minimal-hop shortest-path cycles is explained in detail in this section. Consider the USA Longhaul network shown in Figure 3.1. A highlighted part of the network is enlarged and shown in Figure 3.2.

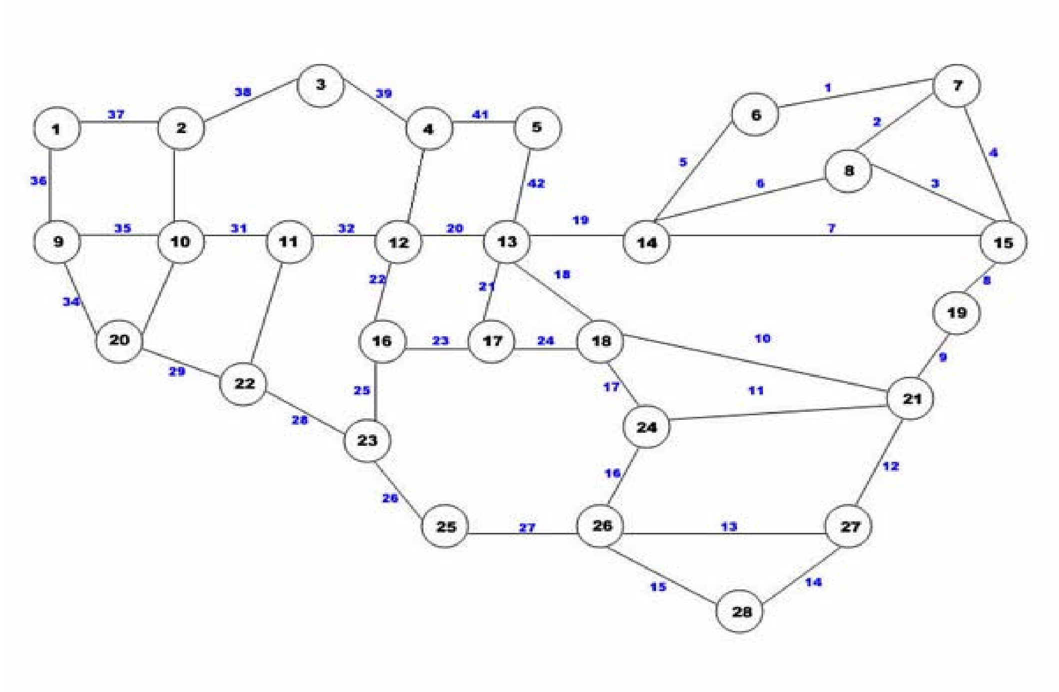


Figure 3.1 The 28 node USA Longhaul Network.

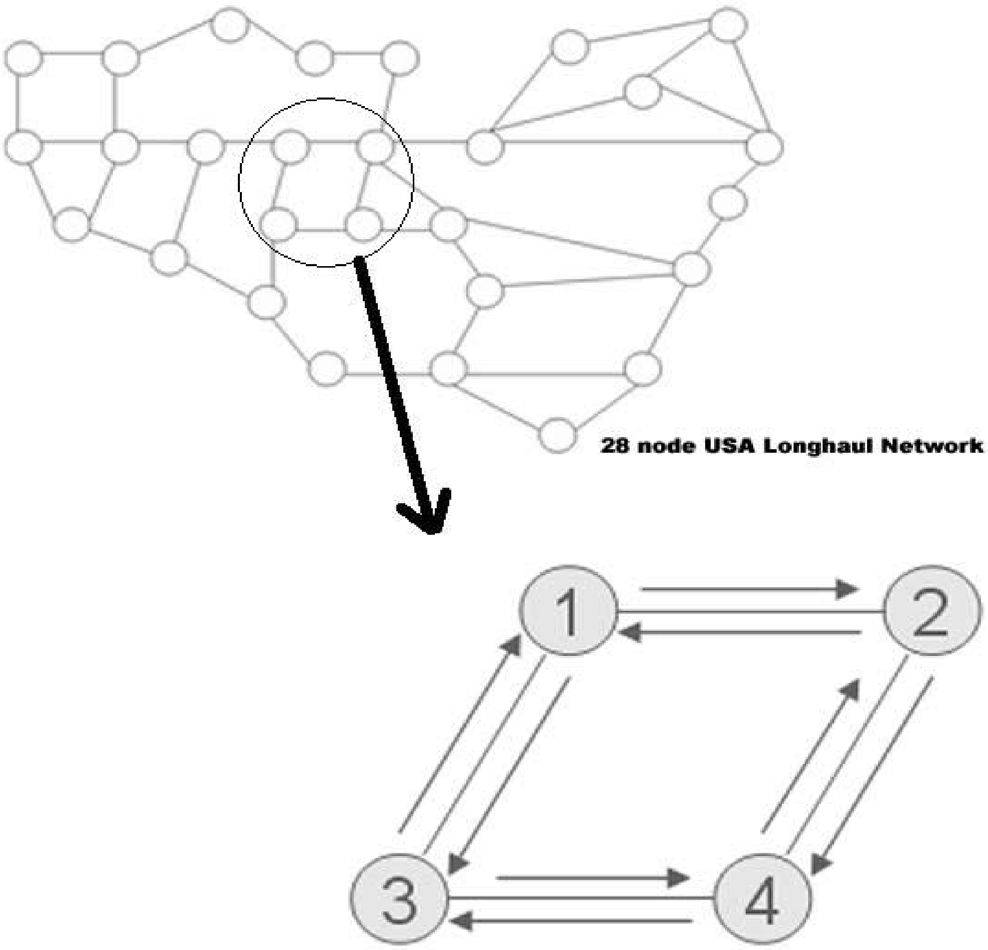


**Figure 3.2 Cycles in the USA Longhaul Network.**

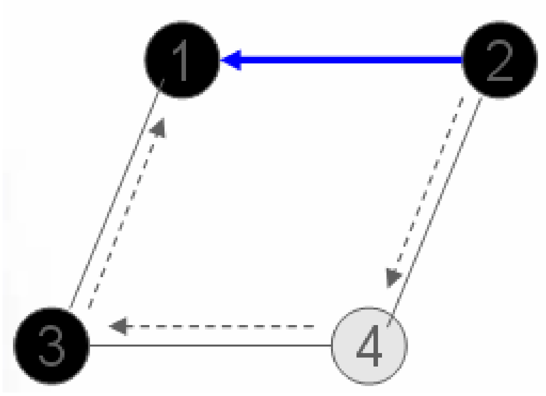


**Figure 3.3 A multicast link as a part of the cycle (Dotted lines indicate backup path in the event of failure of the 1 → 2 link).**

As mentioned earlier all the links in the network are assumed to be bidirectional. Nodes 1, 2, 3 and 4 form a cycle. Because links are assumed to be bidirectional, two unidirectional cycles are formed, i.e., 1 → 2 → 4 → 3 → 1 in the clockwise direction and 1 → 3 → 4 → 2 → 1 in the anticlockwise direction. We utilize these cycles to provide backup paths. Let us now assume that link 2 → 1 is part of the multicast session and that data is being transferred from node 2 to node 1. Such a situation is depicted in Figure 3.3. Link 2 → 1 belongs to the cycle 1 → 3 → 4 → 2 → 1. Keeping 2 → 1 intact and changing the direction of the remaining part of the cycle provides a backup path 2 → 4 → 3 → 1 in the event of failure of link 2 → 1. So if link 2 → 1 fails, the path 2 → 4 → 3 → 1 can be used to deliver data to node 1. This is the main idea behind the scheme proposed in this thesis. We consider each link of the multicast session, one at a time and identify the cycle it belongs to. By doing so, we have identified the backup path in the event of failure of the multicast link.

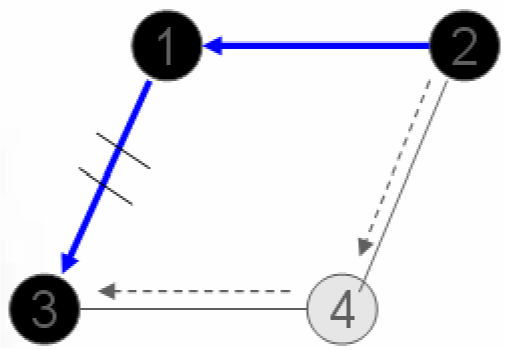


**Figure 3.4 Resource sharing among backup paths in a cycle.**

However, a cycle need not have only one multicast link in it. Consider the situation shown in Figure 3.4. Here link 1 → 3 is also a part of the multicast session. So if link 1 → 3 fails, the path 1 → 2 → 4 → 3 is used as the backup path. Hence, the backup path for link 1 → 3 uses part of the backup path meant for link 2 → 1. This is referred to in this thesis as resource sharing. Due to this feature, we need not compute another cycle for multicast link 1 → 3. This resource sharing technique has a drawback, however. If both links, 2 → 1 and 1 → 3, fail at the same time, this cycle will not be able to deliver power to node 1. Hence, not all multiple link failures can be handled through this scheme.

### 3.4.1 Formation of Cycles in MC Session

The cycles for our algorithm are found in a very simple way. It involves removing the link that belongs to the multicast session from the graph and calculating the shortest path between the end nodes of that multicast link. In order to identify the cycle associated with link 2 → 1 in Figure 3.3, the link is removed from the graph and the shortest path (in terms of number of hops) between nodes 1 and 2 is found. The shortest path thus obtained may include other links in the multicast session. This shortest path put together with the removed link form a cycle that can be used in both the clockwise and the anticlockwise directions. Cycles are allowed to be totally independent of each other or they may overlap with each other.

## 3.5 Algorithm 1 – Recovering from Multiple Link Failures

Algorithm 1 deals with combating multiple link failures in a multicast session. The idea behind the algorithm is to consider one link of the multicast session at a time and for each such link a minimal-hop cycle is calculated. This is done until all the links in the MC session have been covered. This constitutes the Cycle Computation Phase of the algorithm. Once these cycles are computed, they are stored for later access during the recovery phase. The input to the Recovery Phase is the link that has failed. The algorithm then probes the cycle pool to identity the cycle associated with the failed link and then extracts the backup path from the cycle. The backup path is then used to recover from the failure.

### 3.5.1 The Algorithm

**Phase 1: Cycle Computation Phase**

**Definitions:** Every link in the MC session has a STATUS variable and CYCLE_ID associated with it. The CYCLE_ID represents the id of the cycle to which the MC Link belongs. The STATUS variable takes 2 values:

0 – The link is yet to be processed, i.e, covered by a cycle,

1 – The link has already been covered.

**Inputs:** A graph G = (V, A); a multicast session: a source for the multicast and a set of destinations.

**Output:** A set of cycles containing the links in the MC session.

```
begin
Set STATUS of all the links in the MC session to 0 and initialize CYCLE_ID to 0
for each link in the multicast session (in BFS order) with STATUS = 0 {
    // Compute cycles
    Get the end nodes of the link
    Remove the link from the network
    Compute the shortest path between the end nodes of the link using Dijkstra's
    algorithm. /* Assume each edge has unit weight */
    If there is more than one shortest path, select the path with the most MC links
```

```
    Store the shortest path as a cycle with the removed MC link attached to the two
    end nodes of the shortest path.
    // Update status of links in the cycle
    for each link in the cycle just formed {
        if ( link = MC link and STATUS != 1 )
            set STATUS  = 1.
            Assign CYCLE_ID to the link.
        }
    Reinsert the removed link into the network.
    Increment CYCLE_ID.
    }
end-begin
```

**Phase 2: Recovery phase**

**Inputs:** Failed link list and cycle set generated in the Cycle computation phase

**Output:** Back-up paths for the failed links

```
begin
for each link in the failed link list {
    Identify the cycle to which the failed link belongs.
    Traverse the cycle from the end nodes of the failed link to identify the node
    receiving the multicast signal that is nearest to the node immediately affected by
    failure
    From the nearest node to the node immediately affected by failure, extract the
    backup path from the cycle.
```

```
6.      }
7.  end-begin
```

**Illustration of Step 4 and Step 5 of the recovery phase:** Let us assume that the link 26 → 25 belonging to Cycle G in Figure 3.6 has failed. The node immediately affected by this failure is node 25 and the nearest node receiving the multicast signal is node 23. Hence the backup path for this failure is 23 → 25.

### 3.5.2. Illustration of Algorithm 1 – Handling Multiple Link Failures

#### 3.5.2.1 Cycle Computation Phase

The example reported here uses the 28 node USA Longhaul network. Figure 3.5 shows the network and the chosen multicast session. The state of the network is summarized as follows.

Source of the multicast session: 13

Destination set: {10, 20, 12, 25, 18, 26, 16, 24, 23}

Color codes: Destination nodes are represented in dark salmon color and non-destination nodes in light gray. MC links are shown in sienna.

The breadth-first search (BFS) order of the links in the multicast session is 13 → 12, 13 → 18, 12 → 11, 12 → 16, 18 → 24, 11 → 10, 16 → 23, 24 → 26, 10 → 20, 26 → 25. The algorithm starts by removing link 13 → 12 and computing the shortest path between nodes 13 and 12. Link 13 → 12 together with the shortest path thus calculated forms Cycle A. The following is the list of cycles formed considering the MC links in BFS order.

**Cycle A:** 12 →16 → 17 → 13 → 12 covering links 13 → 12 and 12 → 16.

**Cycle B:** 18 → 17 → 13 → 18 covering link 13 → 18.

**Cycle C:** 11 → 22 → 23 → 16 → 12 → 11 covering links 11 → 22 and link 16 → 23. (Link 12 → 16 has already been covered in Cycle A)

**Cycle D:** 24 → 21 → 18 → 24 covering link 18 → 24.

**Cycle E:** 10 → 20 → 22 → 11 → 10 covering links 11 → 10 and 10 → 20.

**Cycle F:** 26 → 27 → 21 → 24 → 26 covering link 24 → 26.

**Cycle G:** 25 → 23 → 16 → 17 → 18 → 24 → 26 → 25 covering link 26 → 25. (Other MC links in this cycle have already been covered in other cycles).

All the cycles together with the MC links they cover are shown in Figure 3.6. These cycles are stored in the cycle pool to be accessed later during the recovery phase of the algorithm.

### 3.5.2.2 Recovery Phase

Consider a situation where the failed links are 13 → 18, 10 → 20 and 26 → 25. In the recovery phase, the algorithm identifies the cycles to which the failed links belong from the cycle pool that was generated in the cycle computation phase of the algorithm. In this case the cycles are B, E and G, respectively. The backup path for link 13 → 18 is in Cycle B and as extracted from Cycle B is 13 → 17 → 18. The backup path for link 10 → 20 is extracted from Cycle E and is 11 → 22 → 20. The node affected due to the failure of link 23 →25 in cycle G is node 25. The node nearest to 25 that is receiving the multicast signal in cycle G is node 23.Hence, the backup path as extracted from the cycle for this failure is 23 → 25

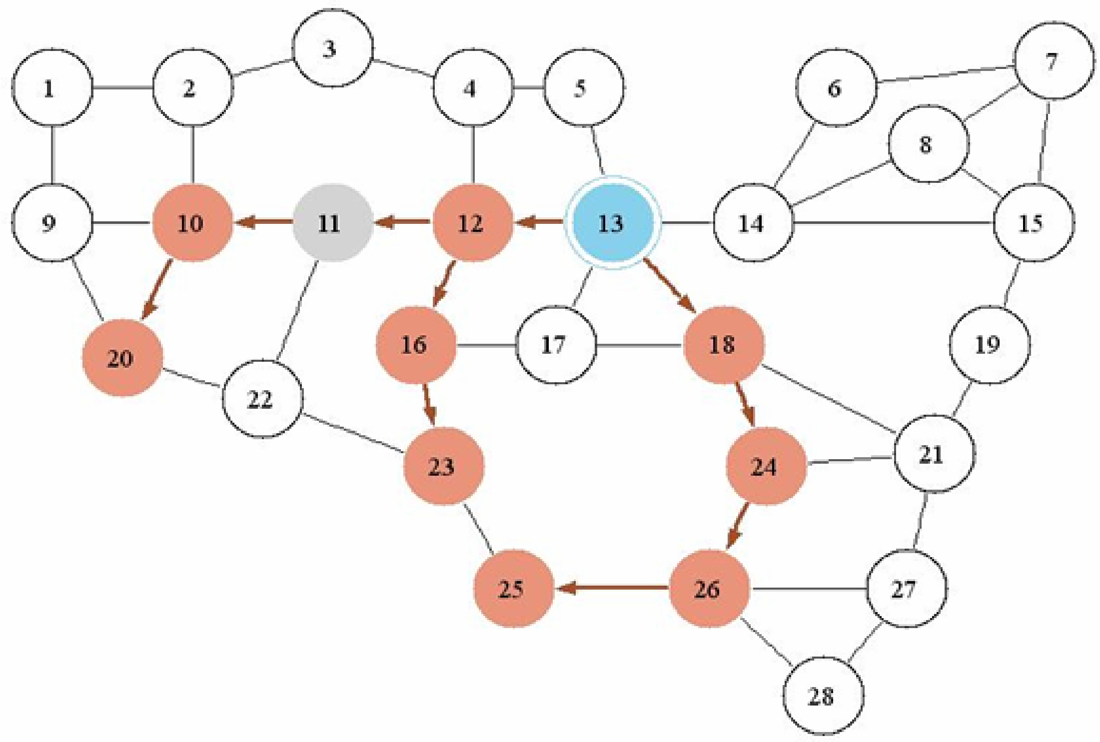


Figure 3.5. USA Longhaul network with an MC session.

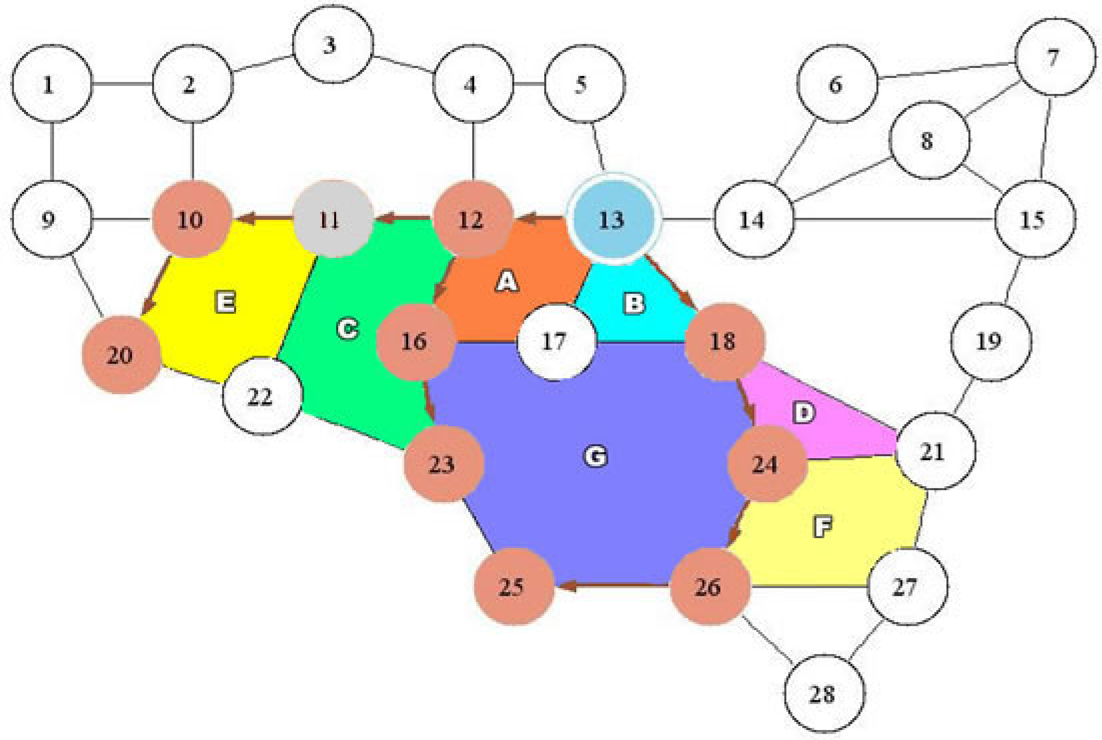


Figure 3.6 Network showing all the cycles of the MC session.

A slight variation of the above algorithm was implemented to study its behavior. Instead of using the shortest path algorithm to form cycles, the depth first search (DFS) algorithm was used. This variation is discussed in the results section.

## 3.6 Algorithm 2 – Recovering from Node Failures

The node failure recovery algorithm, which we call Algorithm 2, is a slight modification of Algorithm 1. This modification allows node failures to be handled successfully by connecting the affected nodes to the source through shortest paths. Algorithm 1 is applied to the unaffected part of the MC session to make it survivable. After the node failure has been effectively handled, the remaining portion of the multicast session, in addition, can support link failures. The fault model and the set of assumptions described for Algorithm 1 hold good for Algorithm 2 as well. In order to reduce the recovery times the faulty node is assumed to have been identified before the cycle computation phase. Hence only link failures are assumed to be dynamic in nature and are known as and when they occur. *Recover by rerouting to the source* concept is used in this algorithm to deal with node failures.

### 3.6.1 The Algorithm

**Phase 1: Cycle Computation Phase**

**Definitions:** Every link in the MC session has a STATUS variable and CYCLE_ID associated with it. The CYCLE_ID represents the id of the cycle to which the MC Link belongs. The STATUS variable takes two values:

0 – The link is yet to be processed, i.e., covered by a cycle,

1 – The link has already been covered.

**Inputs:** A graph G = (V, A); a multicast session: a source for the multicast and a set of destinations; and the faulty node.

**Output:** A set of cycles containing the links in the MC session.

```
1. begin
2. Remove the faulty node from the graph and the multicast session. Let the new MC session be called the modified MC session.
3. Identify all the children of the faulty node that belong to the MC session and place them in the list of affected nodes.
4. for each node in the list of affected nodes {
       Calculate the shortest path from the source of the multicast to the affected node.
   }
5. Merge the shortest paths calculated and the modified MC session to form the new MC session.
6. Run Algorithm 1 on the modified MC session.
7. end-begin.
```

**Phase 2: Recovery Phase**

This phase is identical to the recovery phase of Algorithm 1.

### 3.6.2 Illustration of Algorithm 2 – Handling Simultaneous Node and Link Failures

The example shown here uses the 28 node USA Longhaul network. Figure 3.4 shows the network along with the multicast session. The state of the network is summarized as follows.

Source of the multicast session: 13

Destination set: {10, 20, 12, 25, 18, 26, 16, 24, 23}

Faulty node: 12

### 3.6.2.1 Cycle Computation Phase

Figure 3.7 shows the state of the network after node 12 has failed. The nodes immediately affected by failure of 12 are indicated by 'X' marks. Since *reroute to source* is used in Algorithm 2 the affected nodes, i.e., nodes 11 and 16 are connected to the source using shortest paths after node 12 is completely removed from the network. The path is shown by arrows in Figure 3.8. Algorithm 2 is applied to the unaffected part of the multicast session and the cycles formed are shown in Figure 3.8.

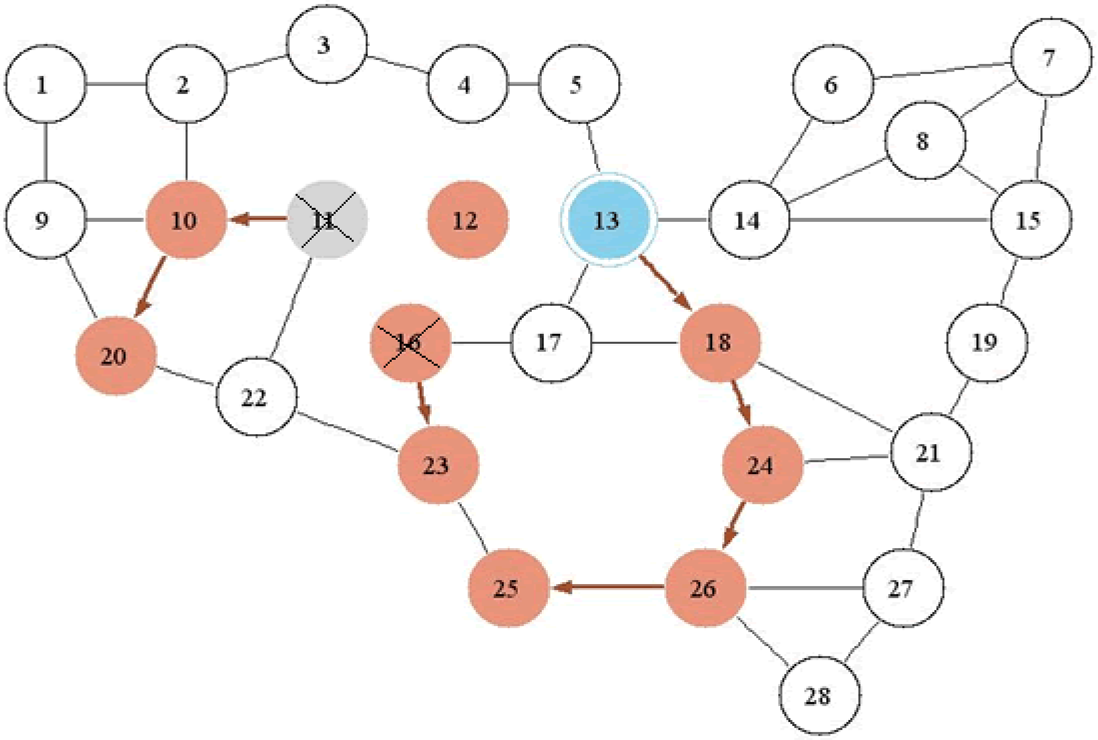


**Figure 3.7 Network showing affected nodes due to failure of node 12.**

### 3.6.2.2 Recovery phase

Failed link: Link 13 → 18.

In the recovery phase, the algorithm identifies the cycles to which the failed links belong, from the cycle pool that was generated in the cycle computation phase of the algorithm. In this case the cycle is Cycle B. The backup path extracted from the cycle is 17 → 18.

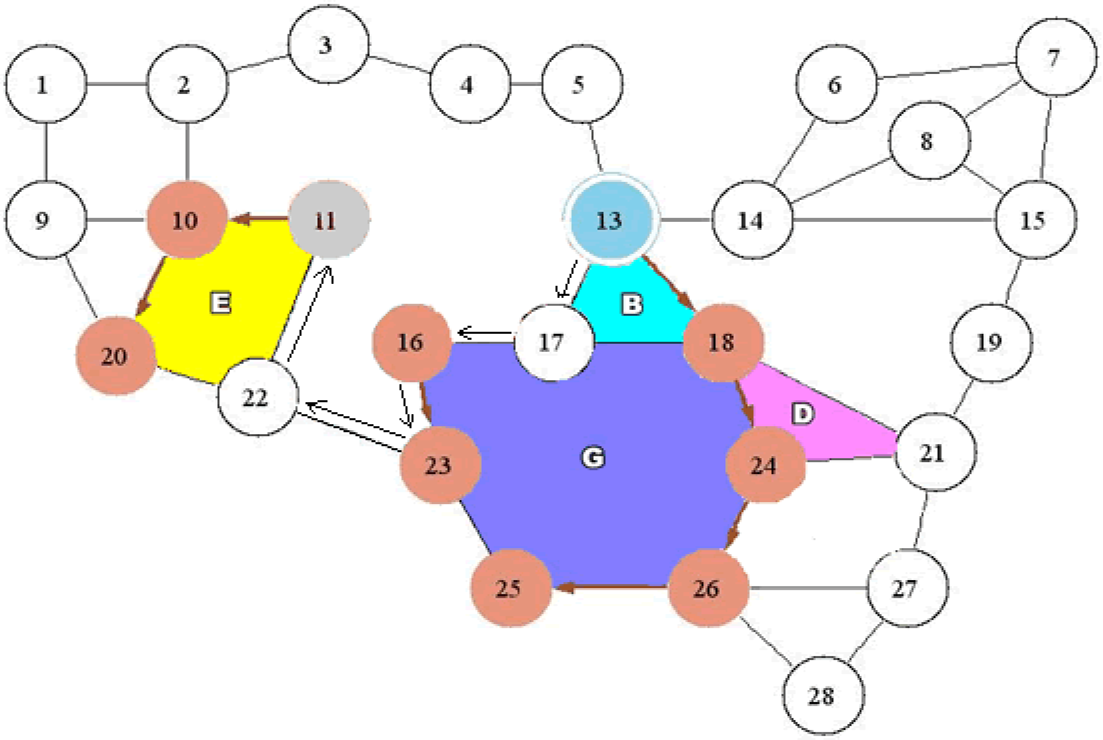


**Figure 3.8 State of the network after connecting affected nodes and forming cycles.**

# Chapter 4

# Results and Discussions

In order to understand the behavior of the algorithms presented in Chapter 3, a simulator was designed with C++ as the implementation language. Simulations were run on an Intel Pentium 4 machine with clock speed of 2.66Ghz and 1GB RAM. The operating system used was Windows XP.

## 4.1 Algorithm Basics

### 4.1.1 Simulator

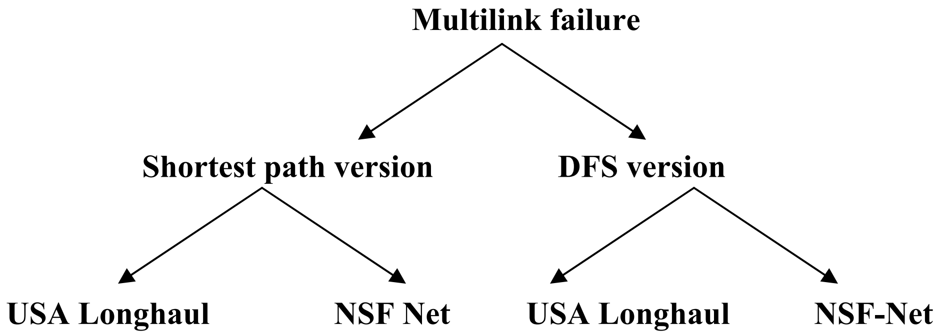


**Figure 4.1 Simulator implementation of the algorithms**

Figure 4.1 gives a brief idea about the implementation of the simulator. Algorithm 1 was implemented as given in Chapter 3. In addition a slight modification of Algorithm 1 was also implemented. Instead of using the shortest path algorithm to establish a path between the end nodes of a deliberately removed MC link in the cycle computation phase of Algorithm 1, a depth first search (DFS) algorithm was used to establish a path.

Although not ideal, formation of cycles using DFS can be used for smaller networks where the probability of multiple link failures is lower.

### 4.1.2 Input Networks

In order to evaluate the performance of the algorithms, they were tested on two different networks. These were:

1. The USA Longhaul network – This network has 28 nodes and is 2-connected. This network is shown in Figure 4.2.

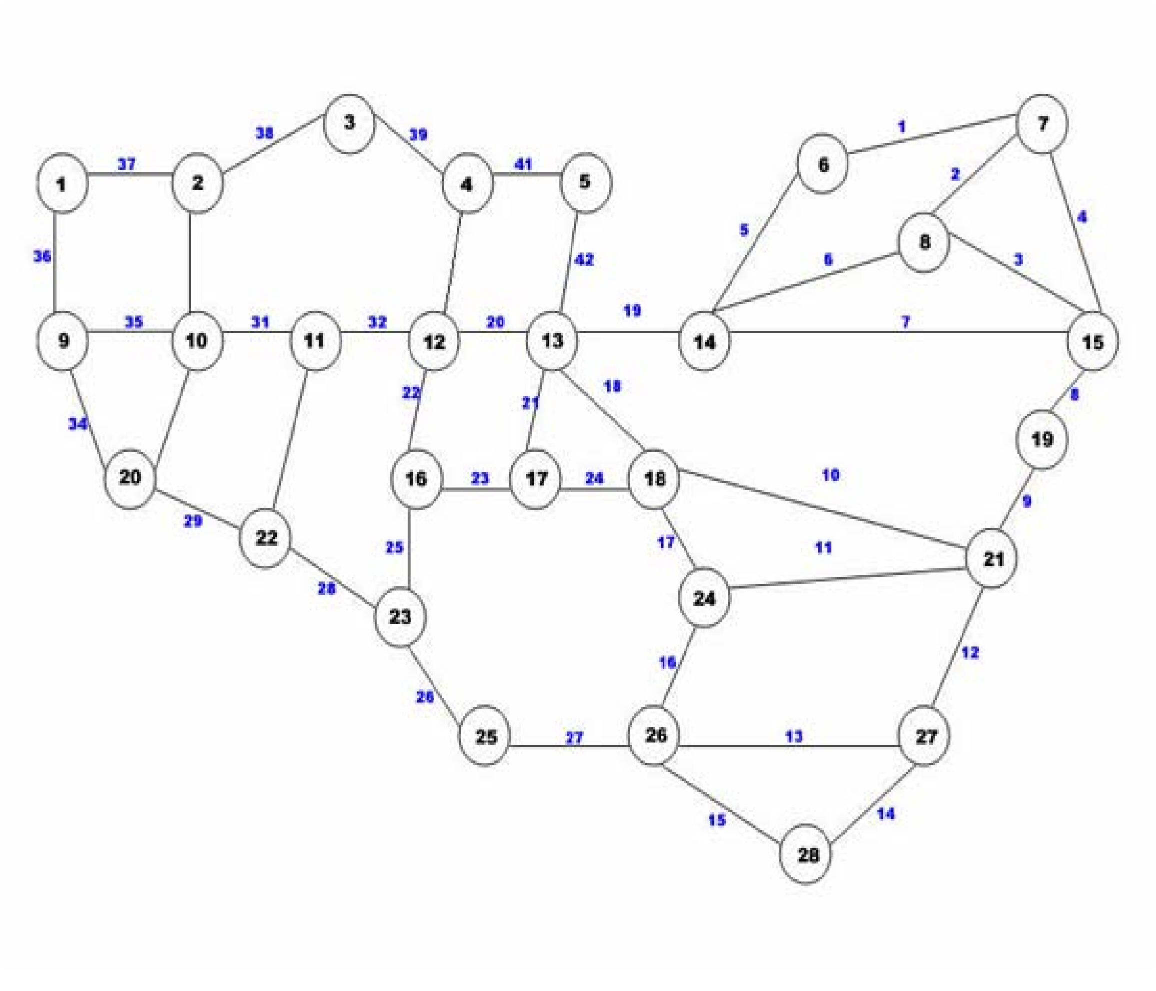


**Figure 4.2 The 28-node USA Longhaul network.**

2. The NSF network – This network has 14 nodes and is 2-connected. It is the most commonly used network for testing algorithms in optical networks. It is shown in Figure 4.3.

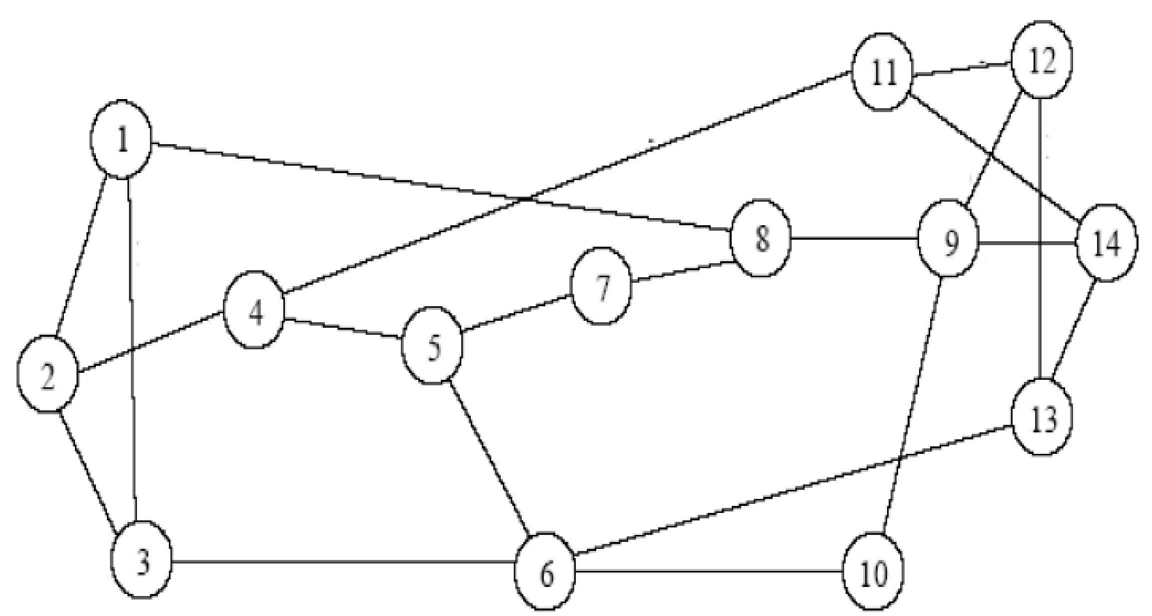


**Figure 4.3 The 14 node NSF network.**

### 4.1.3 Input Multicast Session

All the Multicast sessions used to test our algorithms were derived from a minimal backtracking algorithm presented in [3]. The work done in [3] considers both the splitting and attenuation losses in an optical network and tries to build power efficient multicast trees such that the power received by each destination node in the MC session is higher than a certain threshold level. The work in [3] uses *backtracking* as a novel approach to improve power delivery to destinations. The minimal backtracking algorithm used to generate our MC sessions was implemented with just the first level of pruning

[3]. The source code for this algorithm was obtained from [23]. Six different multicast sessions of varying sizes were used to test our algorithms.

### 4.1.4 Performance metrics

The following parameters were considered in order to evaluate the performance of both algorithms.

a. **Computation time** – Although computation time is dependent upon various external physical factors like processor speed, available memory, etc., it gives an estimate of the time complexity of the algorithms. In our case it represents the time required to compute all the cycles for a given MC session, i.e., the time required to execute the cycle computation phase of our algorithms.

b. **Response time** – Given a link failure, response time includes the time required to probe the cycle pool generated in the cycle computation phase of the algorithm, identify the cycle to which the failed link belongs, get the backup path and update the routing information in the nodes along the backup path.

c. **Average number of cycles** – This parameter represents the average number of cycles formed in an MC session. The number of cycles is indicative of the maximum number of simultaneous multiple link failures that can be supported in an ideal case.

d. **Average number of links** – This parameter gives an idea about the network usage of the algorithm, i.e., the average number of additional links used in backup path for recovery purposes.

e. **Recovery success probability –** Not all simultaneous multiple link failures can be tolerated by Algorithm 1. For example, if two links that are mutually dependent on each other for recovery and belonging to the same cycle fail at the same time, then their simultaneous failure cannot be tolerated. This parameter is indicative of the probability with which a given set of simultaneous link failures can be successfully tolerated.

f. **Percentage change in average power delivered –** This parameter gives an idea of the percentage change in the average power received by the nodes in the MC tree after it has recovered from failures compared to the original power received (before any failures).

The above parameters are measured versus **MC session size**. MC session size can be viewed in two different ways. Since our algorithms deal with processing one link in the MC session at a time, MC session size in terms of links can be viewed as the number of links in the MC session. In terms of nodes, it represents the number of nodes in the MC session (inclusive of both destination and non-destination nodes) other than the source. For all our plots, we considered six different multicast sessions of varying sizes.

## 4.2 Algorithm 1 Results

### 4.2.1 Average Computation Time

As mentioned earlier the computation time gives an idea about the time complexity of the cycle computation phase of Algorithm 1. It refers to the execution time of the cycle computation phase. Given an MC session, it indicates the time required to form cycles that cover all the links in the MC session. The plots shown in Figure 4.4

represent the average computation time. The computation times for both the minimal hop shortest path version and the DFS version of Algorithm 1 are studied by varying the MC session size. The values are plotted for both networks i.e. USA Longhaul and NSF Net.

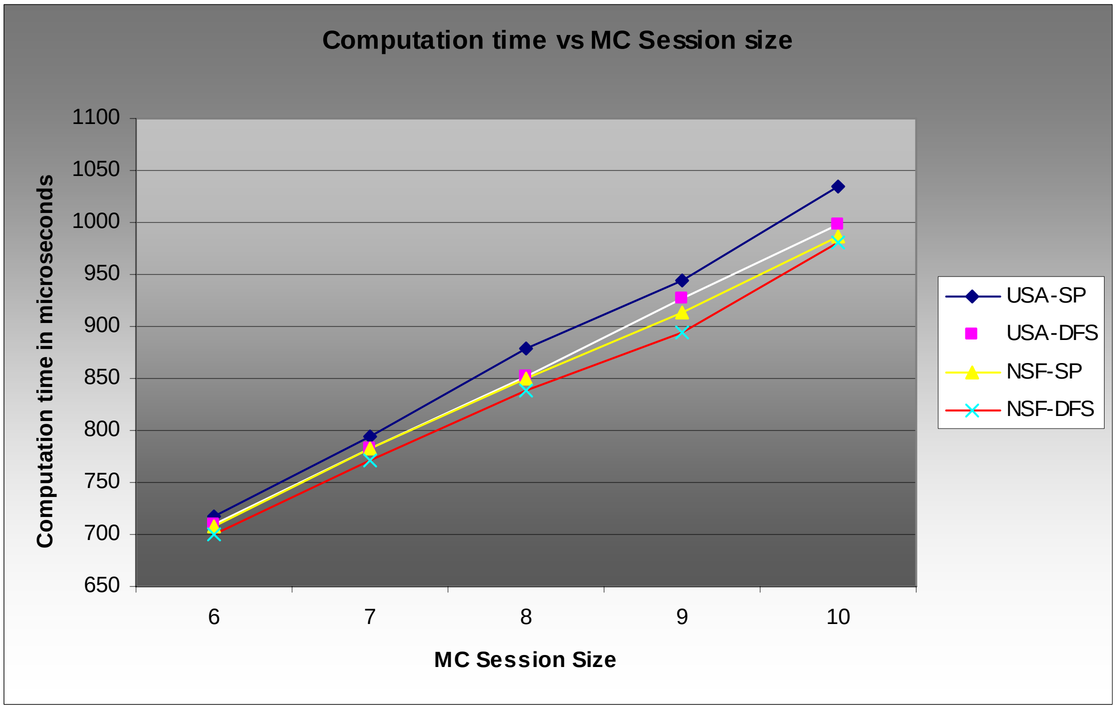


**Figure 4.4 Average computation times vs. MC session size.**

There are two important things to observe from Figure 4.4. Although it is very obvious that the computation time increases as more links and nodes are added to the MC session, it can be seen that the increase is not very rapid or steep. Also from the graph we can see that the DFS version of the algorithm takes less time to execute than its minimal hop shortest path counterpart. This maybe because the depth-first search algorithm is known to execute faster than the Dijkstra's algorithm and the shortest path version of Algorithm 1 has the additional responsibility of resolving conflicts between multiple shortest paths. There are no multiple paths in the DFS version. All the values presented in the graph are of the order of microseconds. Considering the fact that ours is a static approach where the cycle computation phase is executed offline, computation times of the order of few hundreds of microseconds are very much acceptable.

### 4.2.2 Average Number of Cycles

The number of cycles is a rough measure of the possible number of simultaneous link failures that can be tolerated by the algorithm. For example if the MC session size is ten and the number of cycles is six, it implies that six cycles were formed during the cycle computation phase of the algorithm and up to six simultaneous link failures can be supported by the algorithm in an ideal case. For our evaluations we used the average number of cycles as a parameter. From the graph it can be seen that there is an increase in the number of cycles formed as the MC session size increases which is a desirable property.

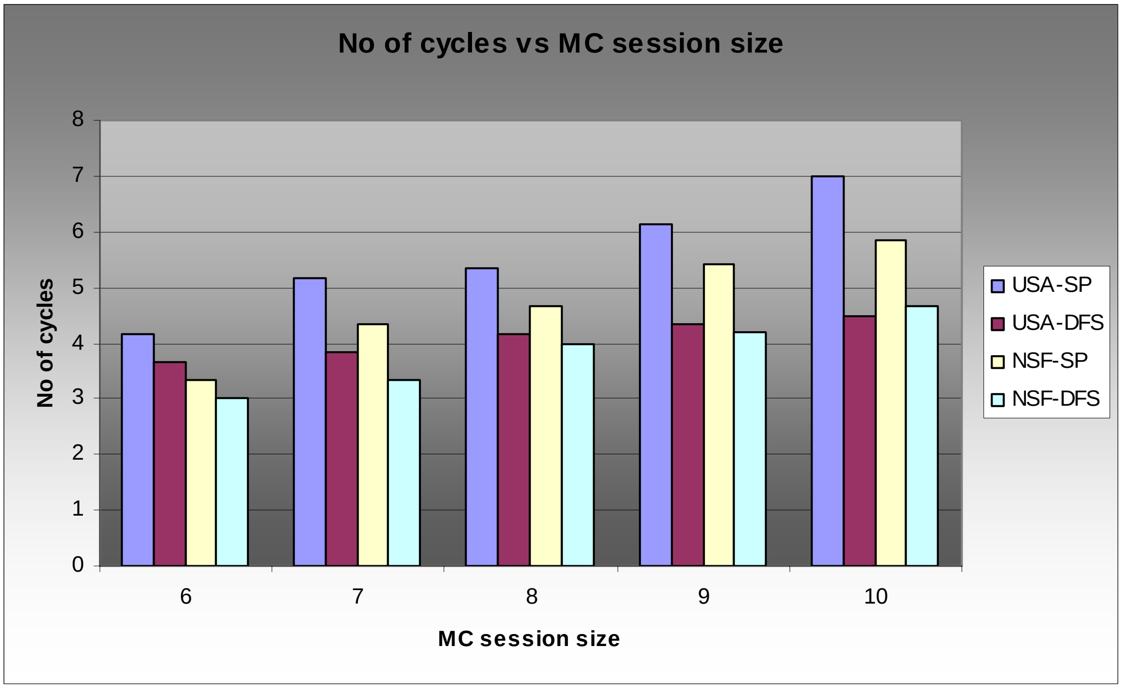


Figure 4.5 Average number of cycles vs. MC session size.

Another important observation is that the number of cycles formed with the DFS version of the algorithm is far less than the number of cycles formed with the minimal hop version of the algorithm. This can be attributed to the fact that the version of the algorithm that uses a depth first search to establish a path between the end nodes of the deliberately failed link in the cycle computation phase would obviously result in a longer path. Hence the the chances of other MC links falling in this path are high. Thus the number of cycles formed is expected to be smaller. Therefore the recovery success probability for the DFS version of the algorithm is smaller than that of minimal hop version. This is discussed in Section 4.2.5.

### 4.2.3 Average Number of Links

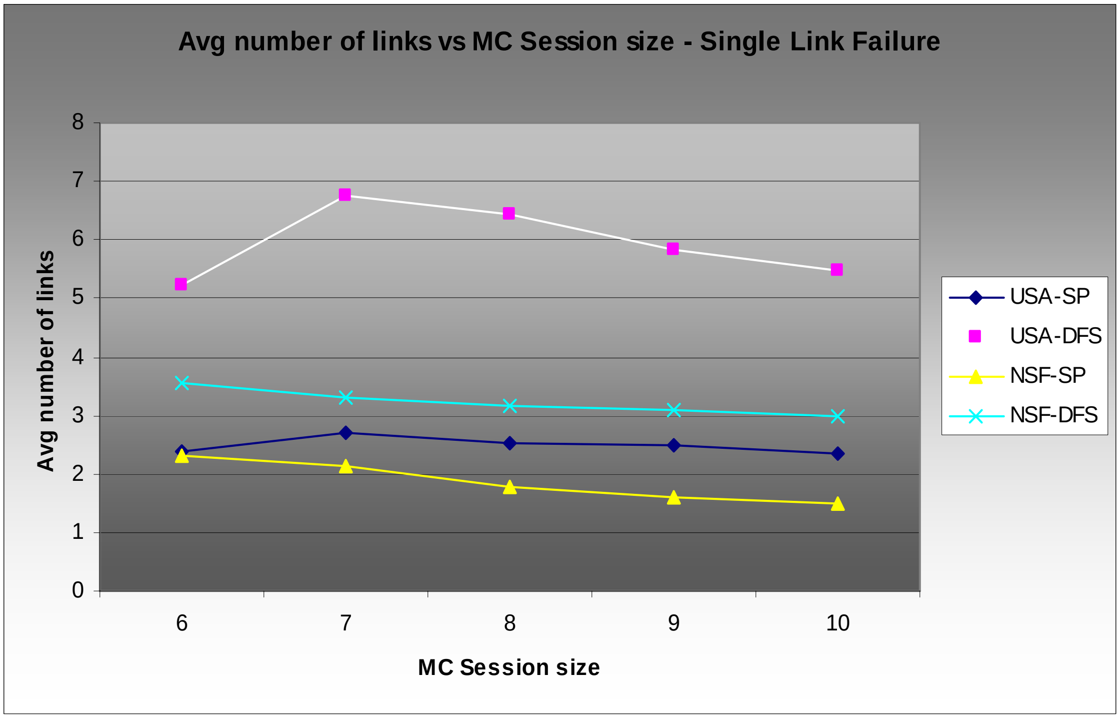


**Figure 4.6(a) Average number of links vs. MC session size for single link failure.**

The average number of links metric gives an idea about the network usage characteristic of the algorithm. It represents the additional number of links that are needed in order to recover from a single link failure, i.e., the number of additional links that are a part of the backup path. While computing this parameter, the links in the back-up path that were already a part of the MC session were not considered. This is because during the establishment of the MC session, resources would have been set aside for this link. If any changes need to be made with respect to this link, only the direction of the flow of data may need to be changed (under certain conditions). However if a new link is added to the MC session as a part of the backup path, new resources need to be allocated (the link needs to be checked for availability, routing information updates need to be made, etc.). Hence, this parameter is an important factor to consider in determining the efficiency of the algorithm since it reflects whether or not network resources are used judiciously and effectively. Four different single link failures were considered to obtain data for each point in the plots of Figure 4.6a.

From the graph it can be clearly seen that the DFS version of the algorithm uses a lot more resources than the shortest path version. This is because the cycles formed in the DFS version of the algorithm are larger in size compared to the cycles in the shortest path version. In other words the number of links that form a cycle in the DFS version of the algorithm is higher than the number of links in a cycle obtained using the shortest path version, hence the difference. Another behavior that can be observed from the graph is that the average number of links tends to decrease as the MC session size increases. This is probably due to the fact that as the MC session size grows, the

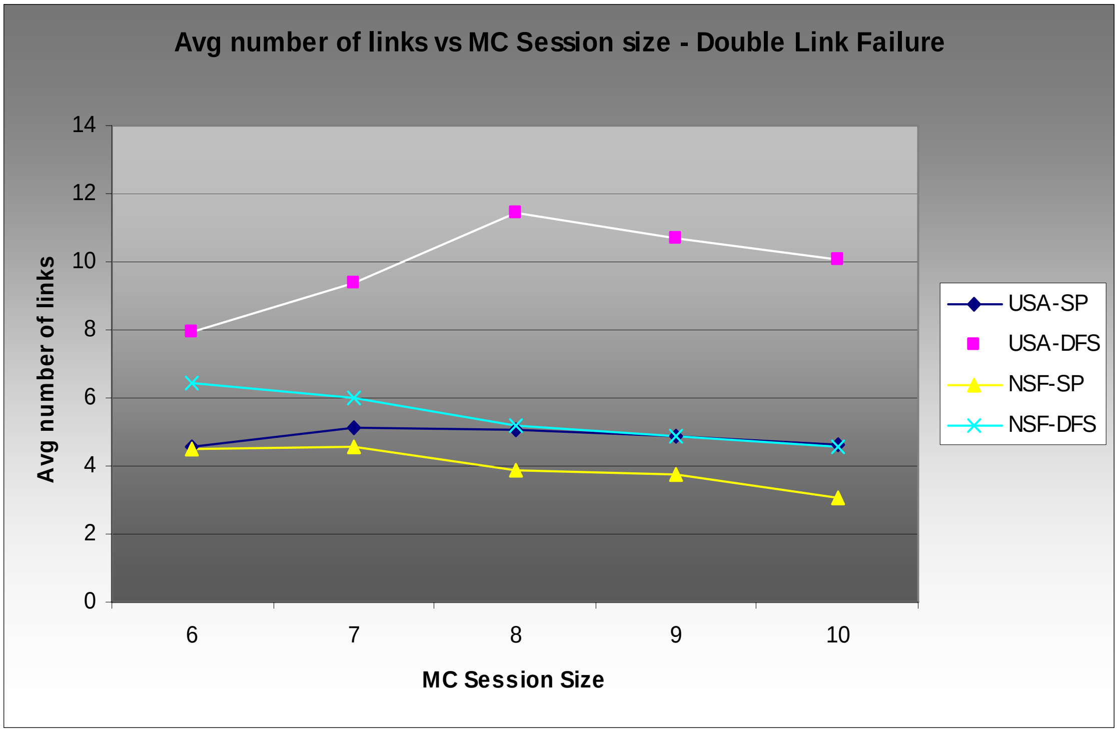


Figure 4.6(b) Average number of links vs. MC session size for double link failure.

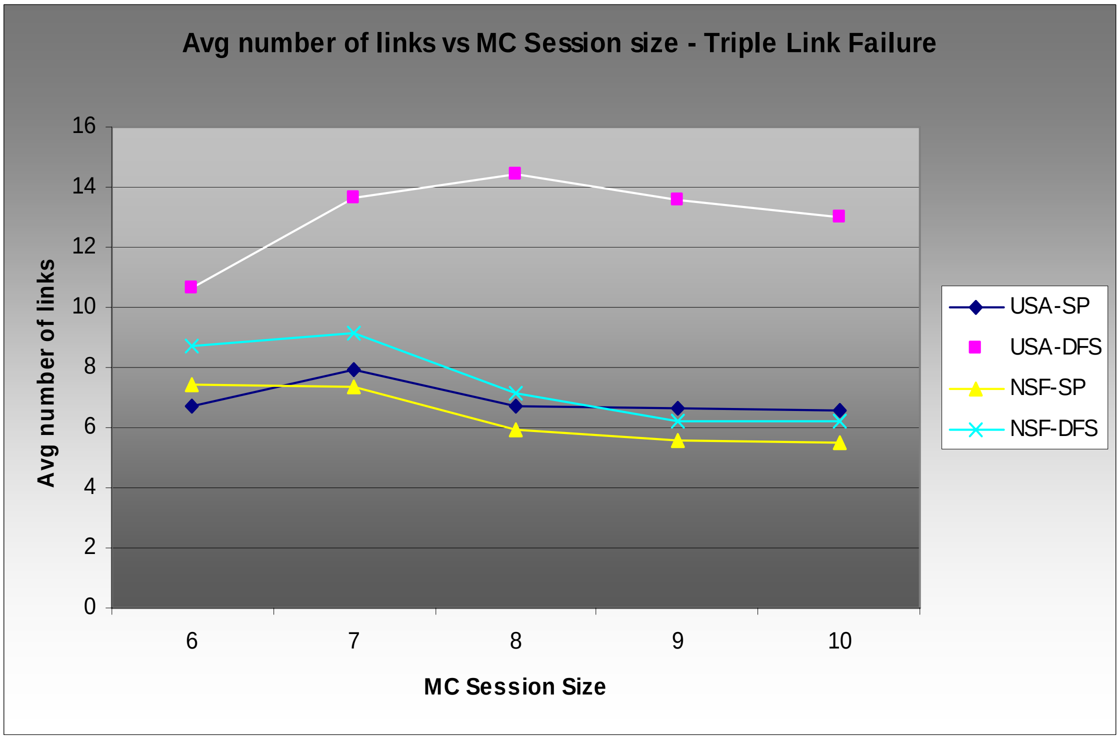


Figure 4.6(c) Average number of links vs. MC session size for triple link failure.

probability of a cycle having more than one link that is a part of the MC session increases. Due to this, the number of additional links that are a part of the backup path used for recovery from a failure within that cycle decreases. The plots for double and triple link failure are shown in Figures 4.6(b) and 4.6(c) respectively.

### 4.2.4.Recovery Times

Recovery time refers to the time required to execute the recovery phase of the algorithm. In other words, given a link failure, this time represents the time required to probe the cycle pool to identify the cycle to which the link belongs, extract the back-up path from the cycle and update routing information of the nodes in the backup path accordingly.

The graphs in Figure 4.7 show the recovery times for one, two and three failures. Four different combinations of double and triple link failures were considered for the plots. As expected, it can be seen from the graphs that recovery times increase as the number of failures increase. It can also be seen that as the MC session size increases the recovery times tend to decrease a bit. This may be because, as the MC session size increases, the number of additional links in the backup path decreases due to which the number of routing information updates tends to decrease. Also it can be observed that the recovery times are larger for the DFS version of the algorithm. This is because of the larger cycle sizes obtained with the DFS version of the algorithm.

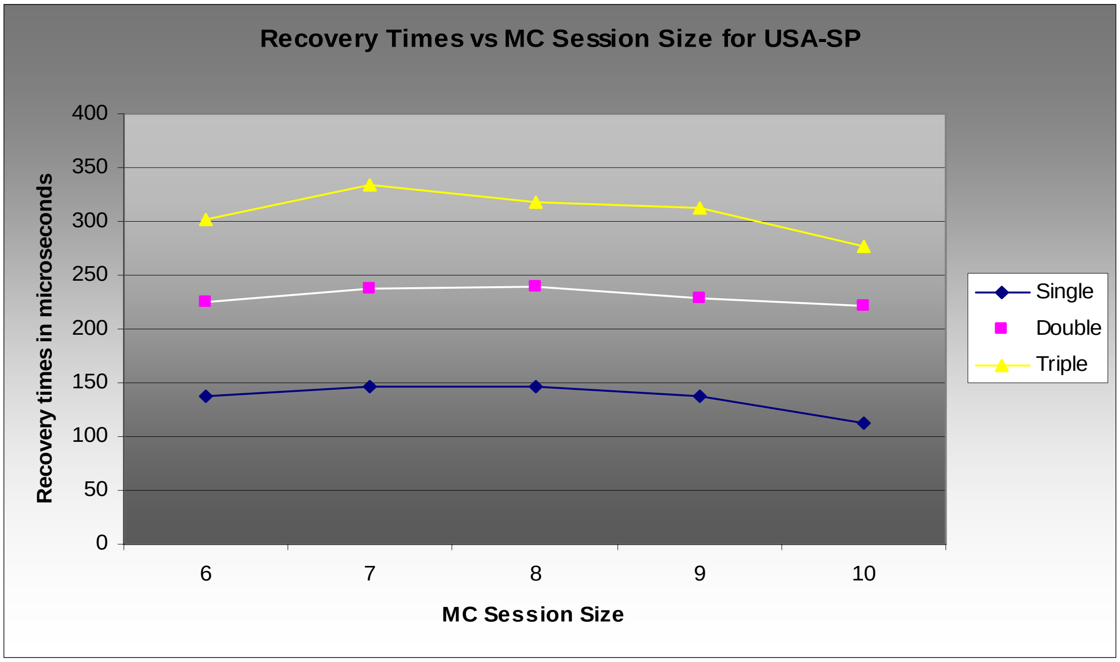


Figure 4.7(a) Recovery times vs. MC session size for USA Longhaul network – Shortest path version.

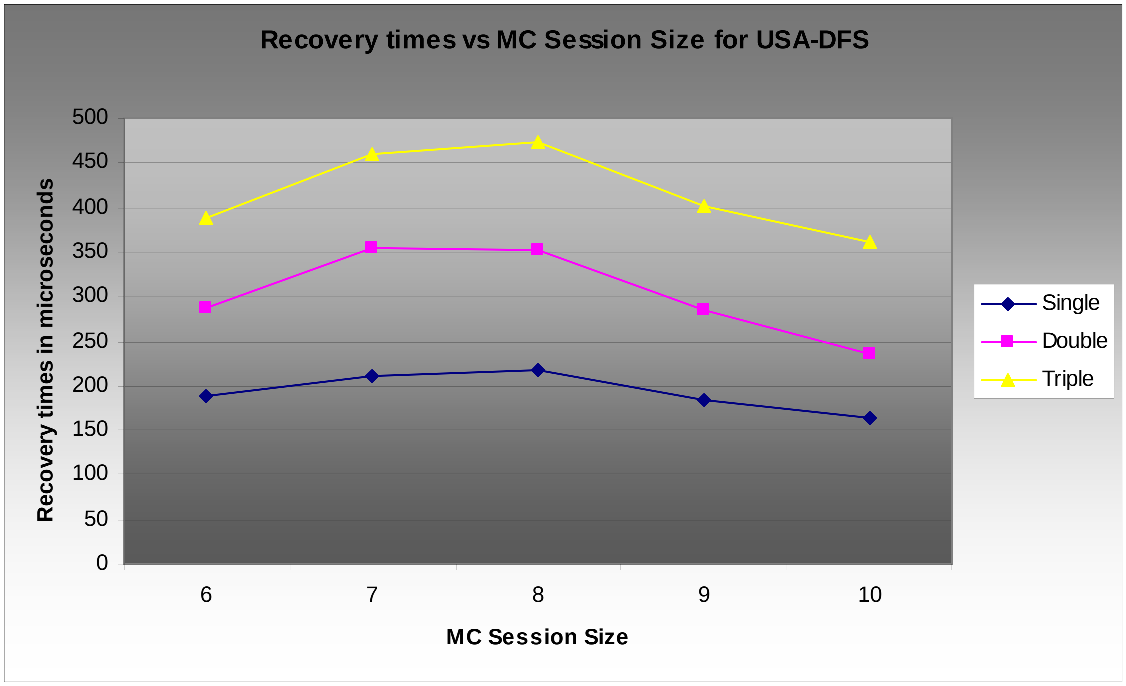


Figure 4.7(b) Recovery times vs. MC session size for USA Longhaul Network – DFS version.

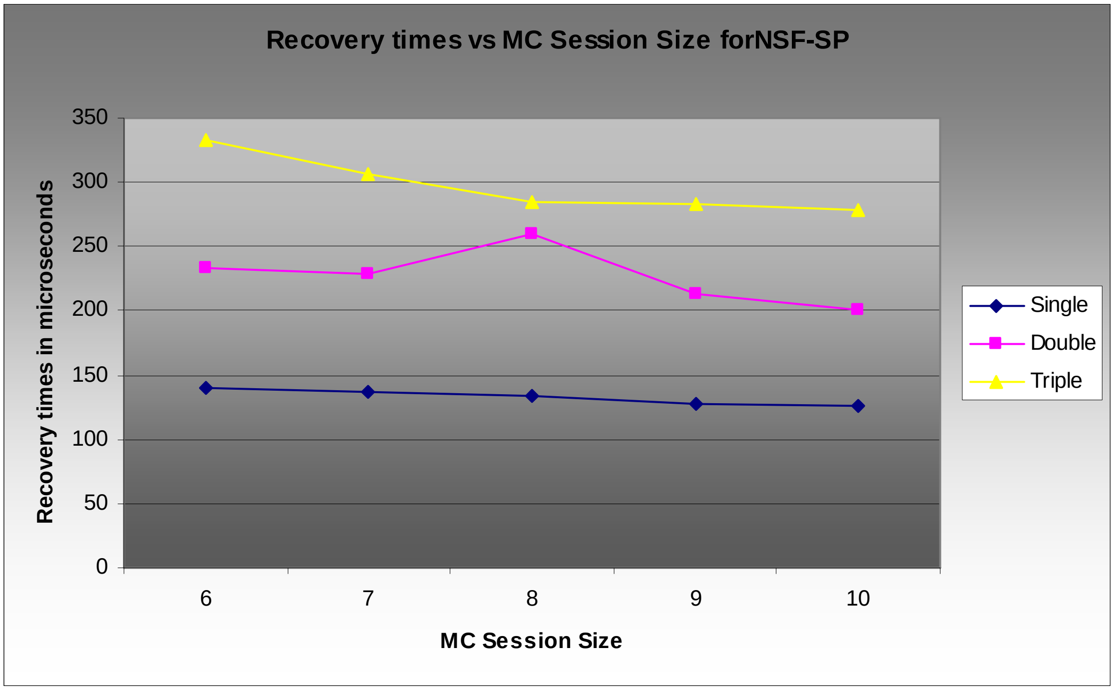


Figure 4.7(c) Recovery times vs. MC session size for NSF network – Shortest path version.

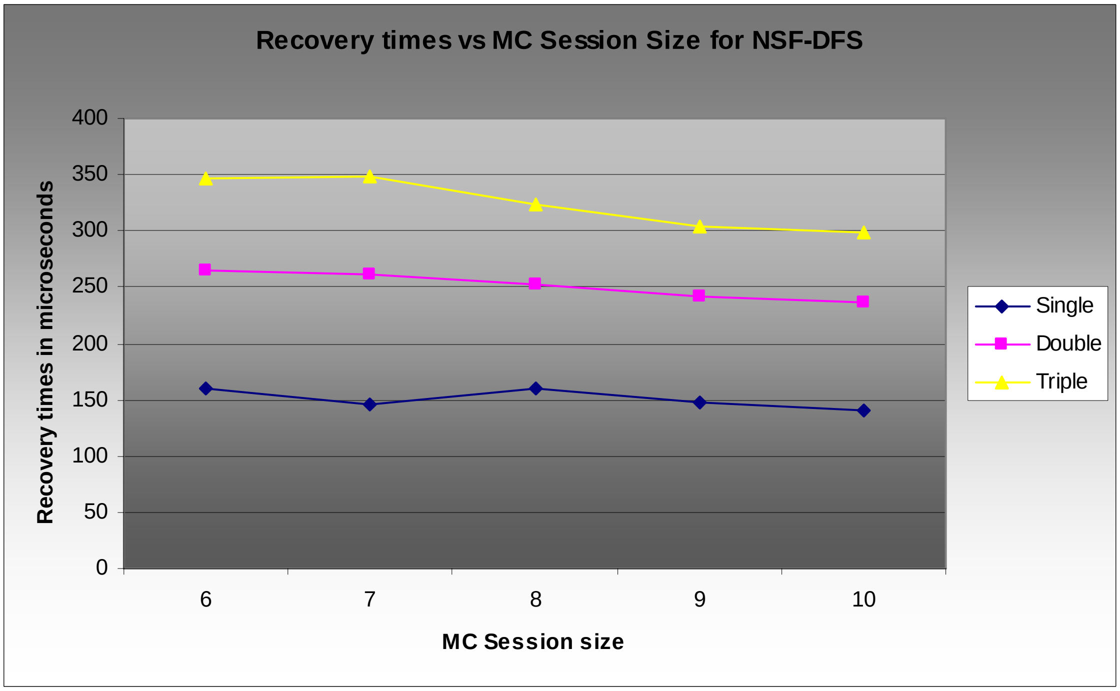


Figure 4.7(d) Recovery times vs. MC session size for NSF network – DFS version.

Due to this the average length of the backup path increases and hence the routing overhead also increases. It can clearly be seen that the minimal hop shortest path version of the algorithm is more efficient and hence should be given preference over the DFS version. Also for the shortest path version the recovery times is of the order of hundreds microseconds. This is highly desirable considering the fact that the recovery phase can be executed in real time. Recovery times for the NSF Network are shown in Figures 4.7c and 4.7d.

### 4.2.5 Recovery Success Probability

Algorithm 1 provides protection against all single link failures in the MC session. However, due to the resource sharing property of the algorithm, it cannot support all simultaneous multiple link failures. For example, if two links that are mutually dependent on each other for recovery and belonging to the same cycle fail at the same time then both failures cannot be tolerated. This is because one failure affects the backup path of the other failure. Hence in this section, we explore the probability with which the heuristic is able to recover from simultaneous two link and three link failures.

Figures 4.8a and 4.8b show the recovery probabilities for USA Longhaul network, while Figures 4.8c and 4.8d show the recovery probabilities for NSF Net. Plots for single link failures are not shown in the graph because the algorithm can handle all single link failures. In other words, the recovery success probability for single link failures for algorithm 1 is 100 percent. All possible combinations of simultaneous two link failures and three link failures were used to study the behavior of the algorithm.

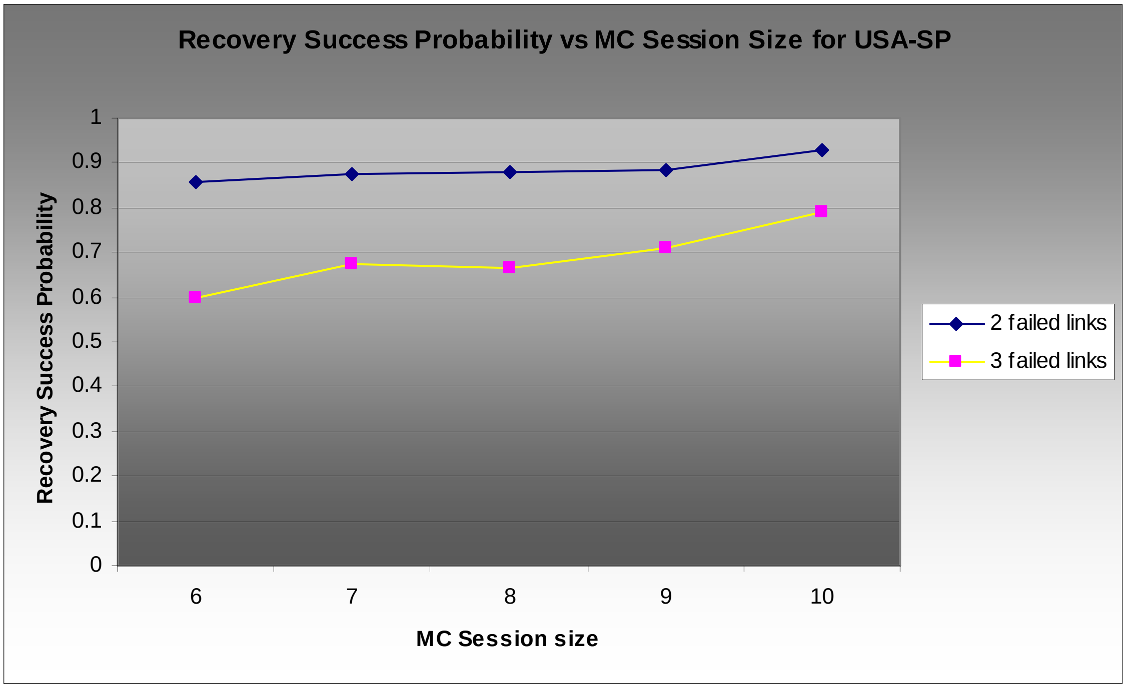


Figure 4.8(a) Recovery success probabilities vs. MC session size for USA Longhaul network – Shortest path version.

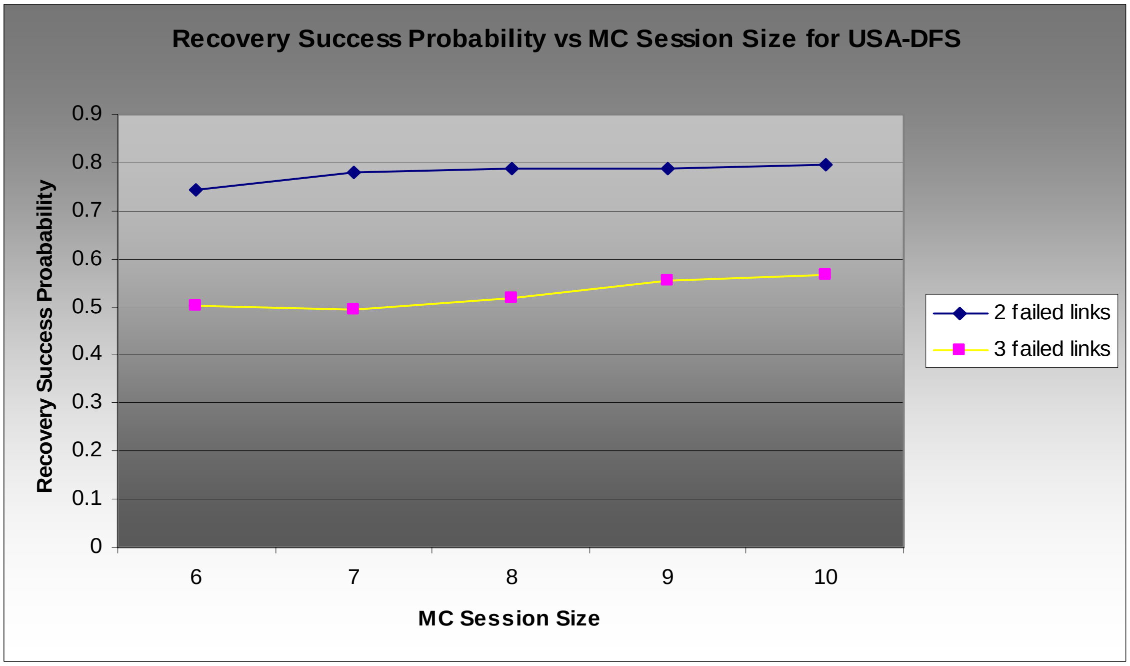


Figure 4.8(b) Recovery success probabilities vs. MC session size for USA Longhaul network – DFS version.

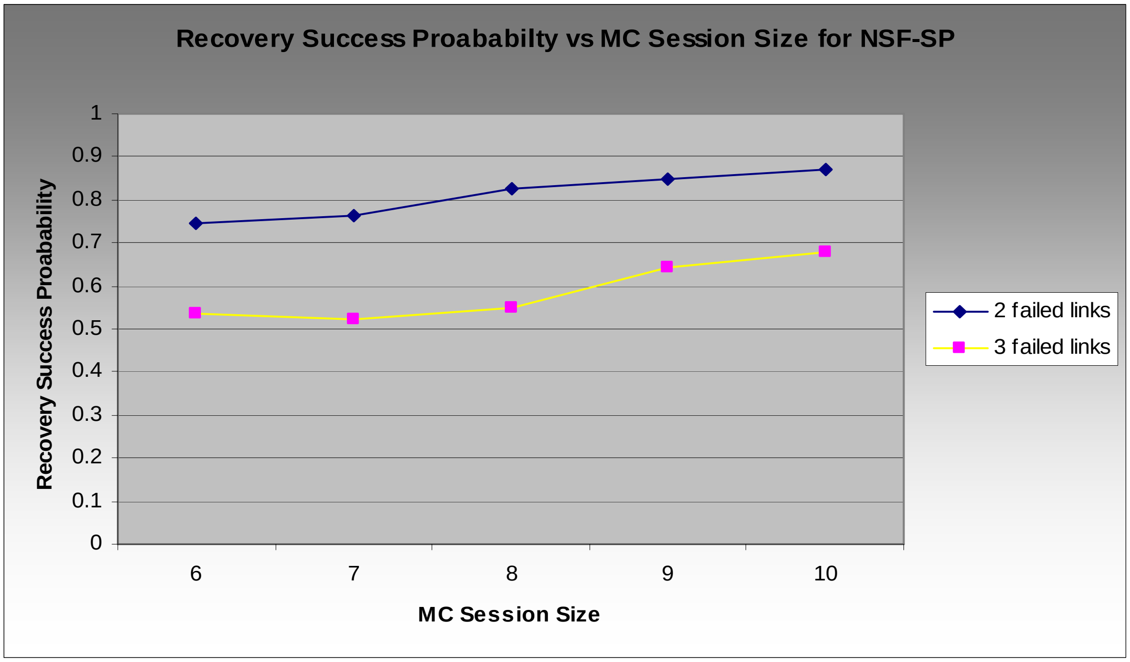


**Figure 4.8(c) Recovery success probabilities vs. MC session size for NSF network – Shortest path version.**

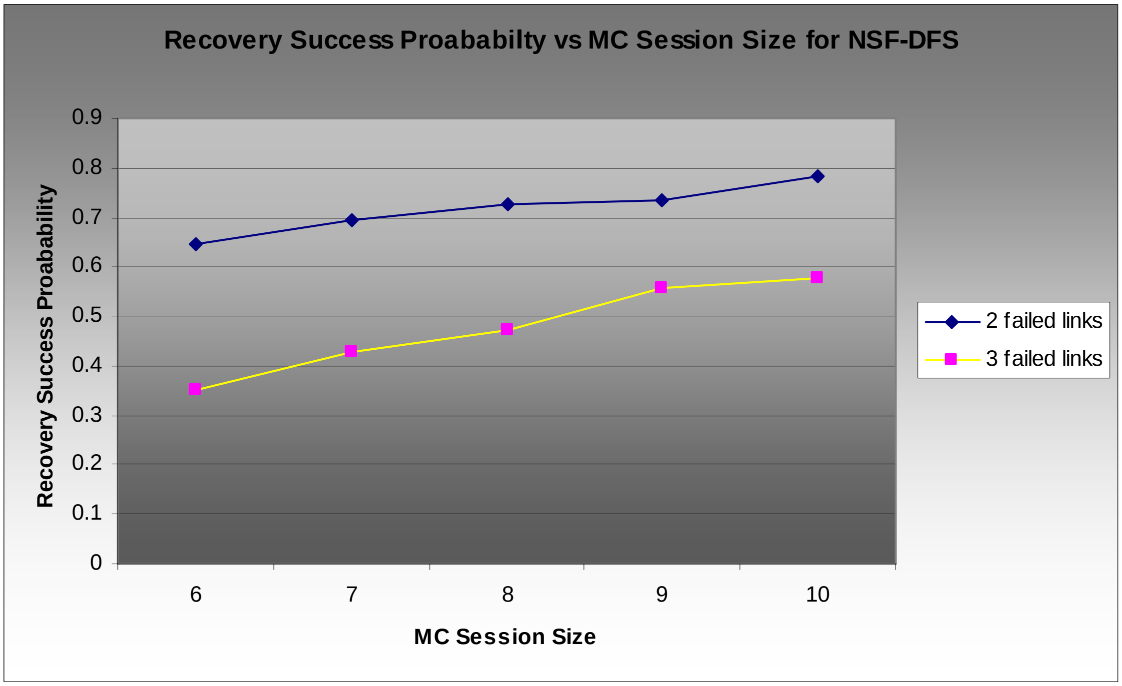


**Figure 4.8(d) Recovery success probabilities vs. MC session size for NSF network – DFS version.**

From the graphs it can be found that the probabilities to recover from two link failures are higher than the probability of recovering from three link failures. Also as the size of the MC session increases the recovery probabilities improve. This may be because of the effect of the number of cycles formed. Figure 4.5 shows that as the size of the MC session increases the number of cycles formed also increases. The higher the number of cycles formed the higher the chances that the failed two (or three) links belonging to different cycles becomes. Hence the probability of surviving these link failures gets higher. Once again the shortest path algorithm outperformed the DFS version of the algorithm mainly because of the reduced number of cycles in the DFS version. For the USA Longhaul network, on an average, the probability of recovering from two link failures was 88.4% and the probability of recovering from three link failures was 68.8% in case of minimal hop algorithm. For DFS these values were 77.9% and 52.7%, respectively.

A minor variation of Algorithm 1 was implemented to see if the recovery probabilities could be improved. The primary factor that determined the recovery probability of the algorithm was the number of cycles formed in the cycle computation phase. In order to increase the number of cycles, we fixed the number of MC links a cycle could contain to two. If a cycle had more than two MC links, separate cycles would be calculated for the additional links in the cycle wherever possible. This modification to Algorithm 1 was tested on the USA Longhaul and NSF networks for the shortest path version of Algorithm 1. The improvements obtained, are shown in dotted lines in the plots of Figures 4.9a and 4.9b.

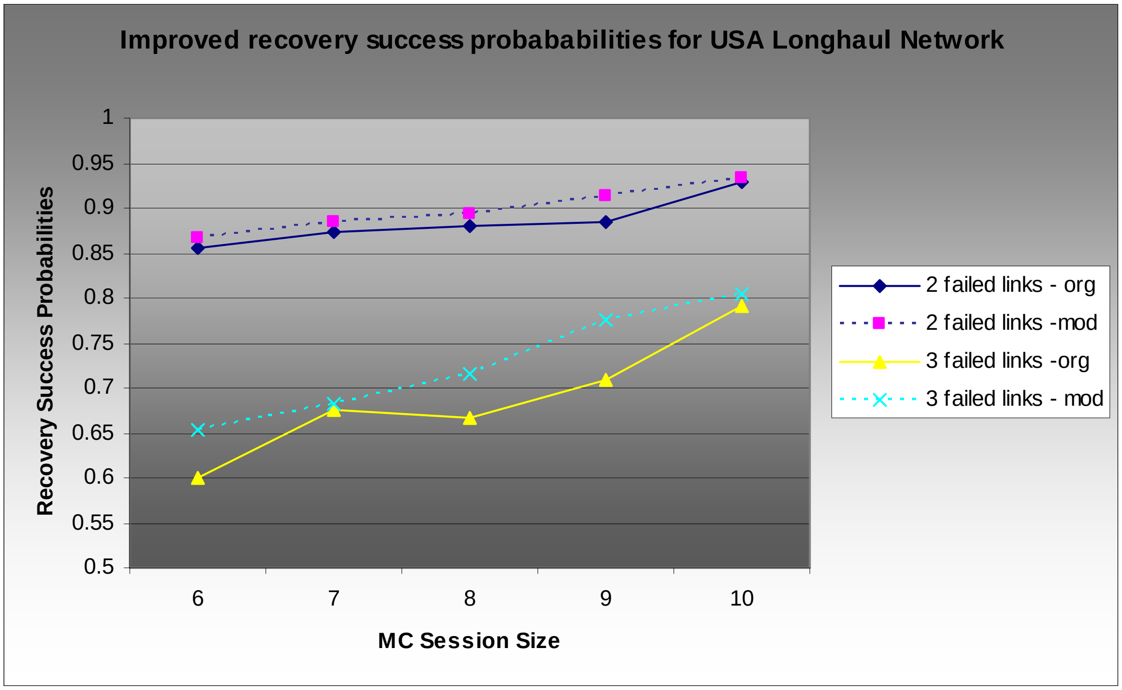


Figure 4.9(a) Improved recovery success probabilities for USA Longhaul network.

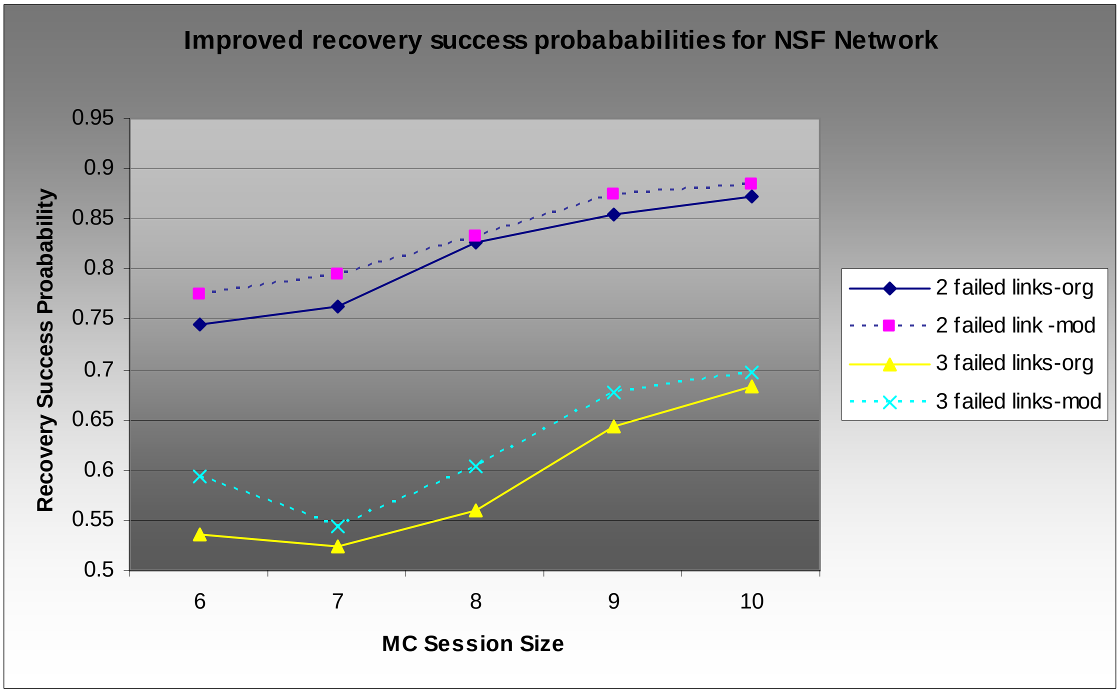


Figure 4.9(b) Improved recovery success probabilities for NSF network.

### 4.2.6 Percentage Change in Average Power

Power losses in an optical network are of two kinds: (a) attenuation losses that result due to propagation of light through the fibers and (b) splitting losses that arise due to splitting of the light at MC capable nodes. Among these two, splitting losses can be very significant. In order to compensate for the losses, most MC capable nodes employ an amplifier at the end of the splitter [17]. The MC sessions for our algorithms were derived from the minimal backtracking algorithm in reference [3]. Percentage change in power refers to the percentage change in average power received in the recovered multicast session compared to the average power received in the original MC session. It indicates the power efficiency of the recovery algorithm. The power at the source was assumed to be unity for all the measurements and the received power was calculated based on the formulae given in reference [27] which take into consideration both the splitting and the attenuation losses in the network. The average values were obtained for as many as three to four different combinations of recoverable double and triple link failures.

Figure 4.10a shows the percentage change in power for the minimal hop shortest path version of Algorithm 1. It can be seen that the change in power increases as the number of failed links increases. This is true because the larger the number of failed links, the larger the number of back-up paths and propagation losses. Another interesting observation is that the percentage change in power decreases as the size of the MC session increases.

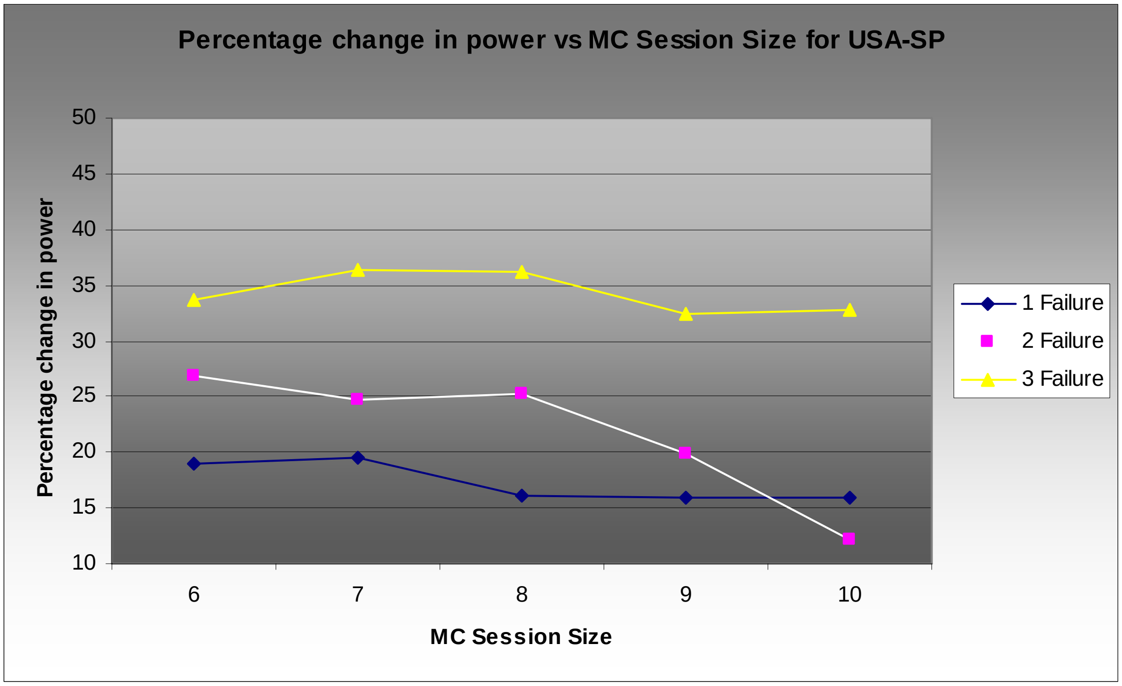


Figure 4.10(a) Percentage change in power vs. MC session size for USA Longhaul network – Shortest path version.

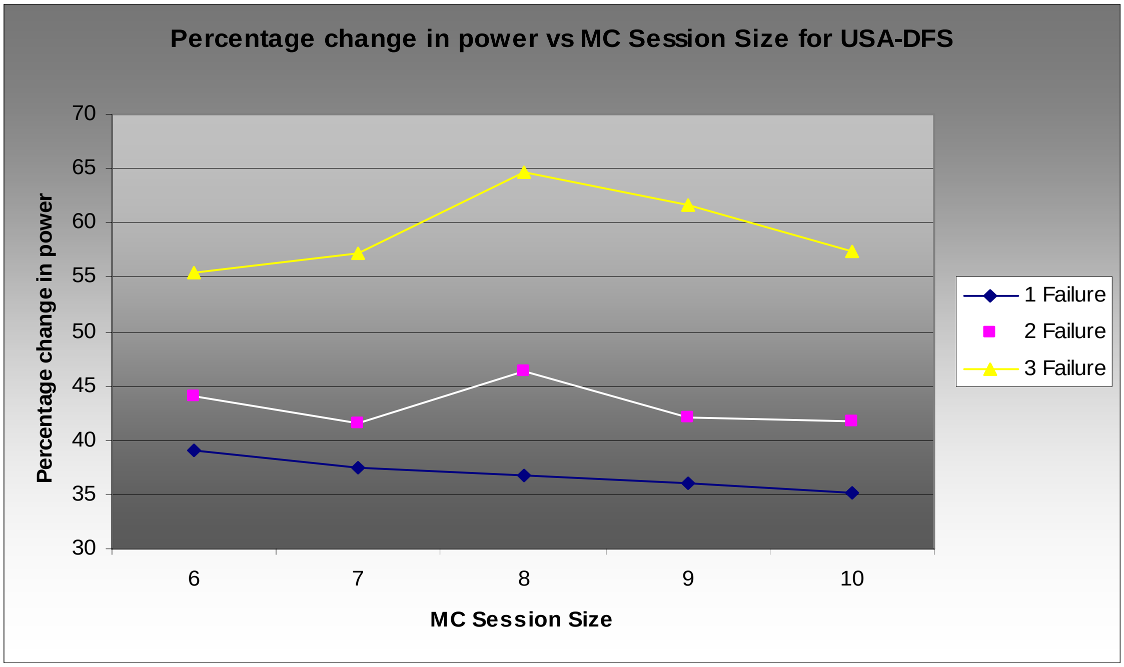


Figure 4.10(b) Percentage change in power vs. MC session size for USA Longhaul network – DFS version.

This could be due to the decrease in the average number of additional links used as the size of the MC session increases. Fewer number of additional links means less propagation losses. Also as the size of the MC session increases, the chances of more than one MC link being present in a cycle become higher. Since the links within the same cycle share a part of the backup path between them if one of them fails, power is delivered to the other nodes during recovery. This may be the reason for the decrease in change in power with the increase in size of the MC session. Here too, the minimal hop shortest path version of the algorithm outperforms the DFS version of the algorithm. The primary reason is again the size of a cycle in the DFS version of the algorithm. Since the number of links is generally higher, the signal has to travel a longer distance. Hence the attenuation losses incurred are more. The behavior of the algorithm for the NSF net is very similar to that for  USA Longhaul network and is shown in Figures 4.9c and 4.9d.

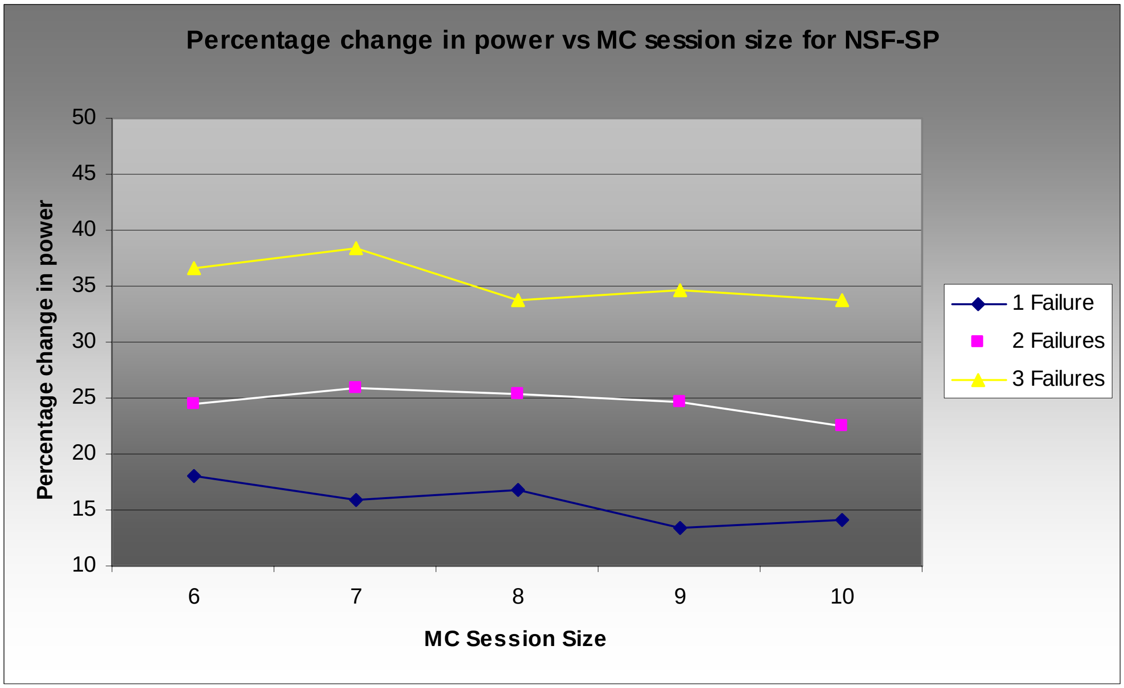


**Figure 4.10(c) Percentage change in power vs. MC session size for NSF network – Shortest path version.**

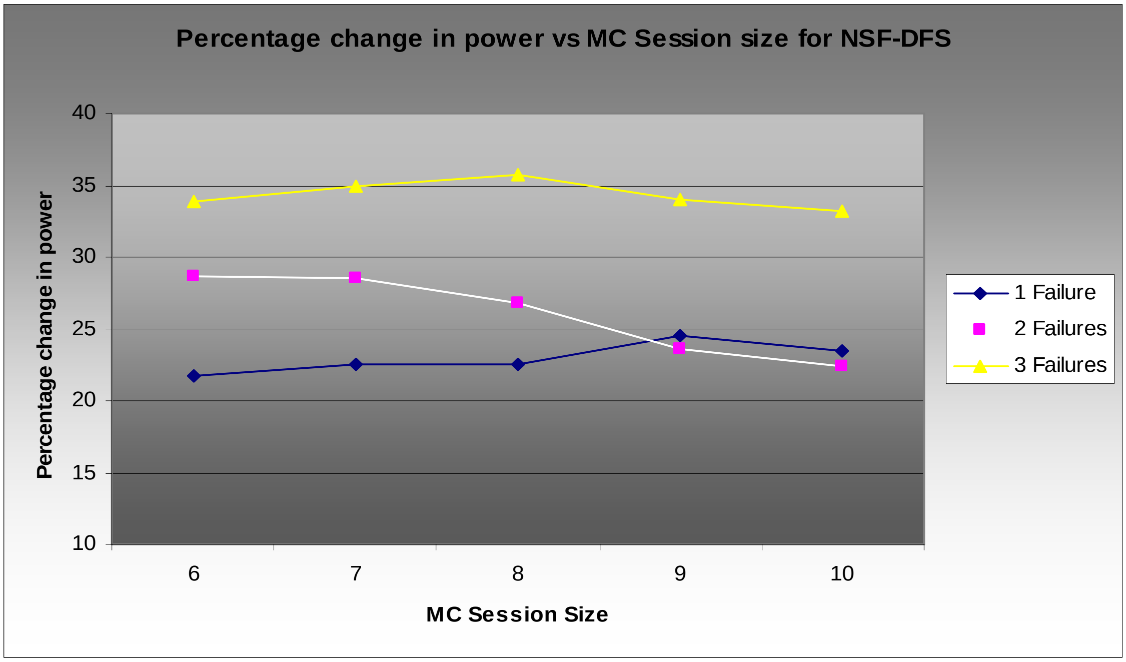


**Figure 4.10(d) Percentage change in power vs. MC session size for NSF network – DFS version.**

## 4.3 Algorithm 2 Results

Algorithm 2 was obtained through a slight modification to Algorithm 1. This algorithm not only tolerates link failures but can also handle node failures. This section studies the behavior of Algorithm 2. From the previous section, it is clear that the minimal hop shortest path version of Algorithm 1 performs better than its DFS counterpart. However, the DFS version of Algorithm 1 has its own advantages. It has shorter computation times and can be used for small networks where the chances of multiple link failures are less. For studying node failures, we consider the shortest path version only. Only one node failure is considered at any time and only those nodes whose failure affects the multicast session downstream are considered. Hence these nodes can either be destination nodes or non-destination nodes that are members of the MC session.

### 4.3.1 Average Computation Time

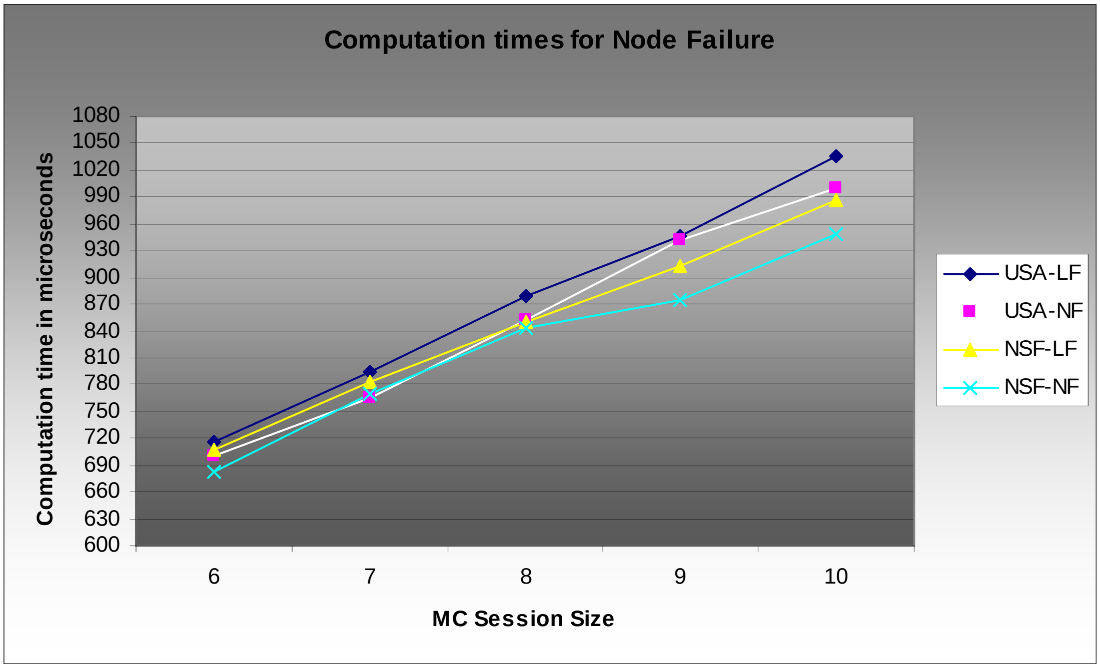


**Figure 4.11 Computation time vs. MC Session size for Algorithm 2**

Figure 4.11 shows the computation time for Algorithm 2 vs. MC session size. It can be seen that the computation times for Algorithm 2 was slightly lower than that for Algorithm 1. On an average, for the MC sessions considered, it was found that a node failure resulted in the failure of two to three links. Hence these two to three links were not considered for shortest path calculations during the cycle computation phase of Algorithm 2. However, the time saved due to this was offset by the time to compute the paths from the source to the affected nodes. This is probably the reason for the difference in computation times. Hence, a node failure did not significantly impact the computation time of the algorithm.

### 4.3.2 Recovery Success Probability

The networks considered here for testing our algorithms are at least 2-connected. The recovery success probability considered in Section 4.2.5 had probability plots for simultaneous double and triple link failures only. This is because Algorithm 1 can successfully handle all single link failures. However, the same case is not true under conditions of node failure in a 2-connected network. The probability plots in this section consider simultaneous node and link failure and provide the probabilities of recovering from all possible single link failures. The table in Figure 4.12 reports the behavior of Algorithm 2 with the USA Longhaul and the NSF networks.

It was found that the probability of surviving a single link failure was higher in NSF network when compared to USA Longhaul. In networks that are 2-connected, under certain circumstances, when a node fails, the failure could leave some nodes with a connectivity of one (i.e., removal of a single link could disconnect the node). If the node is part of the MC session, then the link associated with this node cannot be recovered if it fails. The number of such nodes is higher in USA Longhaul network compared to NSF

| Probabilities | USA | NSF |
|---|---|---|
| 6 | 0.812 | 0.936 |
| 7 | 0.837 | 0.957 |
| 8 | 0.94 | 0.925 |
| 9 | 0.923 | 0.967 |
| 10 | 0.916 | 0.989 |
| Avg | **0.8856** | **0.9548** |

**Figure 4.12 Recovery success probabilities for USA Longhaul and NSF networks for Algorithm 2.**

network. This may be the reason for the difference in the recovery success probability between the two networks. Although this graph doesn't show recovery probabilities for two and three link failures, that doesn't imply that the algorithm cannot support them. They are not plotted here because the probability of a node failure together with two or three link failures is very slim. A rough measure of the recovery success chances for two and three link failures together with a single node failure can be obtained from the graph shown in Figure 4.13. The average number of cycles parameter is not studied for its own sake but to quantify the possible number of link failures that an MC session subjected to Algorithm 2 can sustain even after a node fails.

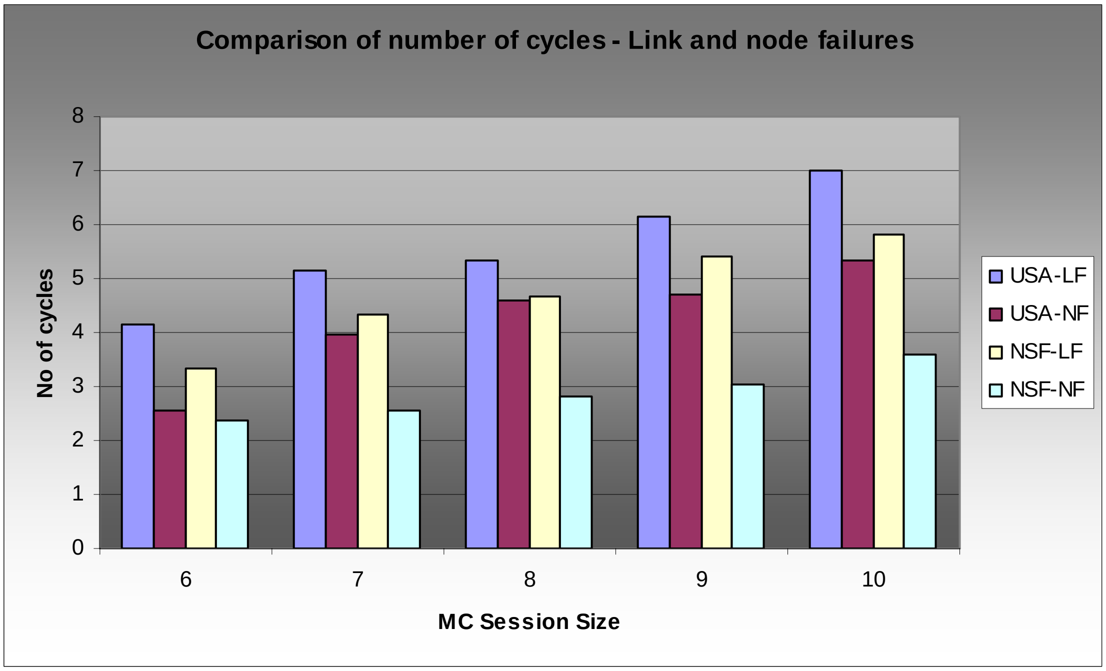


**Figure 4.13 Average number of cycles vs. MC session size for Algorithm 2.**

The above graph compares the number of cycles for Algorithm 1 and Algorithm 2 for both networks.

### 4.3.3 Percentage Change in Average Power

The graph shown in Figure 4.14 compares the percentage change in average power delivered using Algorithm 1 and Algorithm 2. It is meant to study the effect of node failure on the power delivered to the MC session. The yellow and the white lines in the graph depict values for the USA Longhaul and the NSF Network, respectively, under conditions of simultaneous node and link failure. The other two plots in the graph are the plots from Algoirthm 1 derived under conditions of single link failure.

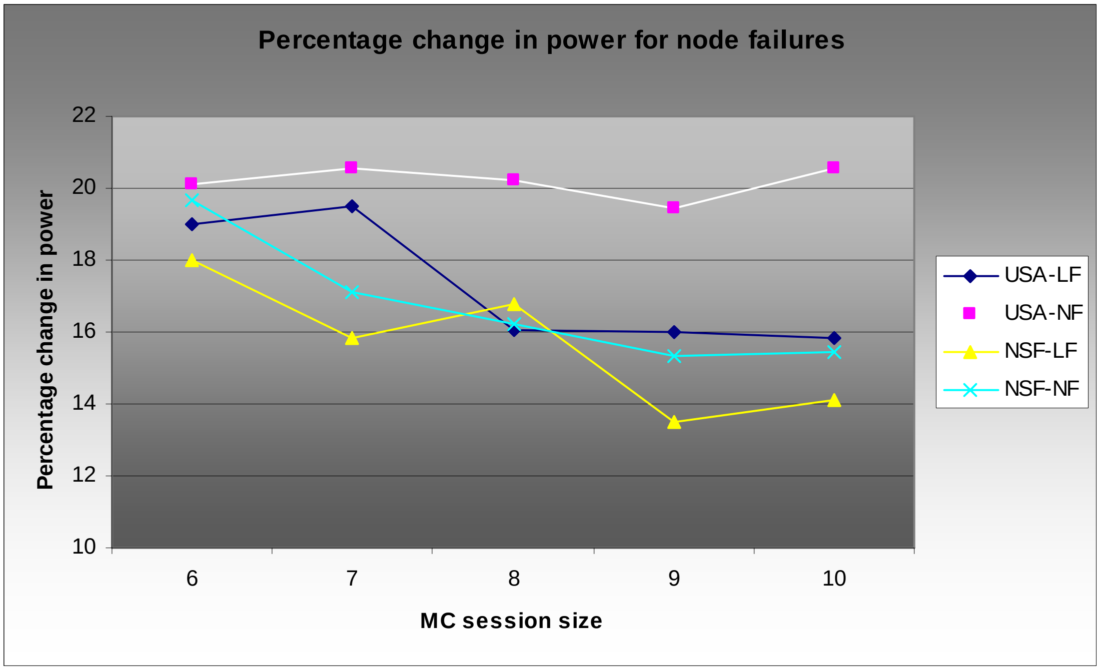


**Figure 4.14 Percentage change in power vs. MC session size for Algorithm 2.**

The formulations given in reference [27] were once again used to obtain these plots. Four different node failures were considered to obtain the points of the graph. The nodes considered were both destination and non-destination nodes provided that their failure affected the rest of the MC session downstream. Three separate single link failures for each node failure were considered for the plots. From Section 4.3.2 it was found that not all single link failures can be tolerated by Algorithm 2. Hence only those single link failures that could be tolerated were the ones that were considered.

It was found from the graph that, although there were separate paths to the affected nodes from the source, the percentage change in power was higher when compared to Algorithm 1. This may be due to two reasons. Since a node failure results in the failure of all the links associated with it, the connectivity of the graph is affected. Hence the new paths established from the source to the affected nodes are longer. Also when the new paths established overlap, as shown in the example for Algorithm 2, splitting losses are induced. Hence, the average power delivered to the MC session under conditions of simultaneous node and link failure is less when compared to the conditions of single link failure only.

# Chapter 5

## Conclusions and Future Enhancements

Component failures such as link failures and node failures are responsible for data losses and network impairments in a WDM optical network. This thesis presented two algorithms to make a multicast session resilient to both link and node failures. The algorithms presented in this thesis use a hybrid approach to tolerate component failures. They can be viewed as a combination of proactive and reactive schemes. The cycle computation phase is executed offline and the recovery phase is executed when a link failure is detected. Algorithm 1 uses the novel concept of minimal hop cycles to tolerate simultaneous multiple link failures in a multicast session. We explore the possibility of resource sharing among backup paths through the use of cycles. Two different approaches for the formation of cycles were presented in the thesis. It was found that the shortest path version of the algorithm performed better than its DFS counterpart. Algorithm 2 considered the case of having a node and a link fail at the same time. Keeping the concept of the cycle formation intact, Algorithm 2 uses the recover by rerouting to source concept to tolerate node failures. The algorithms were tested on two different networks and performance was studied under several constraints.

The performance metrics used to study the behavior of our algorithms were computation time, network usage, recovery success probability, percentage change in power delivered to the destinations, response times and average number of cycles. Although computation times for the DFS version of Algorithm 1 were shorter, it was found that the shortest path version of Algorithm 1 was more power efficient, used less network resources and had a higher probability of recovering from simultaneous multiple link failures when compared to the DFS version. Although not explicitly mentioned, Algorithm 1 can tolerate single link failures with 100 percent probability. The shortest path version of Algorithm 1 showed very fast recovery times in the event of failures. Also with Algorithm 2 it was found that the NSF network had a higher probability of recovering from single link failures than the USA Longhaul network under conditions of node failure.

Future enhancements to these algorithms may include, but are not limited to, improving the time complexity of the algorithms and better methods to form cycles. We considered a fully splitting network for our algorithms but the same concept can be extended to sparse split networks as well.

# References

[1]. M. Ali and J. S. Deogun., "Power-Efficient Design of Multicast Wavelength-Routed Networks," IEE J. Select. Areas Commun, vol. 18, no. 10, Oct. 2000, pp. 1852-1862.

[2]. R. J. Bates, “Basic Fiberoptics Technologies”, Publication: New York McGraw-Hill Professional, 2001.

[3]. 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.

[4]. X. Cao, V. Anand, Y. Xiong and C. Qiao, “A study of waveband switching with multilayer multigranular optical cross-connects”, Selected Areas in Communications, IEEE Journal on Volume 21, Issue 7, Sept. 2003 Page(s):1081 – 1095.

[5]. X. Cao, V. Anand and C. Qiao, “Multilayer versus single-layer optical cross-connect architectures for waveband switching”, INFOCOM 2004. Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies Volume 3, 2004 Page(s):1830 - 1840 vol.3.

[6]. X. Cao, V. Anand and C. Qiao, “Waveband switching in optical networks”, Communications Magazine, IEEE Volume 41, Issue 4, April 2003 Page(s):105-112.

[7]. http://www.cisco.com/univercd/cc/td/doc/cisintwk/ito_doc/fddi.pdf

[8]. X. Chu and B. Li, “Dynamic routing and wavelength assignment in the presence of wavelength conversion for all-optical networks”, IEEE/ACM Transactions on Networking, Volume 13, Issue 3, June 2005 Page(s):704 – 715.

[9]. R. C. Elsenpeter, and T. J Velte, “Optical Networking: A Beginner's Guide.”, Publication: New York McGraw-Hill Professional, 2002.

[10]. M. T. Fredrick and A. K. Somani, “A single-fault recovery strategy for optical networks using subgraph routing,” in Proceedings of the 7th IFIP Working Conference on Optical Network Design and Modelling, Budapest, Hungary, February 2003, pp. 327–346.

[11]. J. He, S.H.G. Chan, D.H.K. Tsang, “Routing and wavelength assignment for WDM multicast networks”, Global Telecommunications Conference, 2001. GLOBECOM '01. IEEE Volume 3, 25-29 Nov. 2001 Page(s):1536 - 1540 vol.3.

[12]. http://www.iec.org/online/tutorials/opt_net/topic12.html

[13]. V. Kompella, J. Pasquale, and G. Polyzos. “Multicast routing for multimedia communication.”, IEEE/ACM Transactions on Networking, 1(3):286--292, June 1993.

[14]. H. C. Lin and C. H. Wang, "A hybrid multicast scheduling algorithm for single-hop WDM networks," in Proc. of IEEE Infocom '01, Anchorage, Alaska, Apr. 2001, pp. 169--178.

[15]. R. Malli, X. Zhang, and C. Qiao, "Benefit of multicasting in all-optical networks," in SPIE Proc. Conf. All-Optical Networking, vol. 2531, Nov. 1998, pp. 209--220.

[16]. G. Mohan and C. Siva Ram Murthy, “Lightpath Restoration in WDM Optical Networks,” IEEE Network Magazine, vol. 14, no. 6, pp. 24-32, November/December 2000.

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

[18]. http://www.nren.nasa.gov/workshop8/pps/13.E.Future_Blumenthal.pdf

[19]. A.E Ozdaglar and D.P Bertsekas, “Routing and wavelength assignment in optical networks”, Networking, IEEE/ACM Transactions on Volume 11, Issue 2, April 2003 Page(s):259 – 272.

[20]. R. Ramaswami and K.N Sivarajan, “Routing and wavelength assignment in all-optical networks”, IEEE/ACM Transactions on Networking, Volume 3, Issue 5, Oct. 1995 Page(s):489 – 500.

[21]. G. Rouskas, “Routing and wavelength assignment in optical WDM networks”, John Proakis, editor, *Wiley Encyclopedia of Telecommunications*. John Wiley & Sons, 2001.

[22]. L. H. Sahasrabuddhe and B. Mukherje, “*Light-trees: Optical multicasting for improved performance in wavelength-routed networks.*”, IEEE Communications, 37(2):67--73, February 1999.

[23]. S. C. Shekhar, “Survivable Multicasting in WDM Optical Networks”, A Master’s thesis, Dept. of Electrical Engineering, Louisiana State University, Baton Rouge, 2004.

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

[25]. N. Sreenath, N. Krishna Mohan Reddy, G. Mohan and C. Siva Ram Murthy, “Virtual Source Based Trees: A New Approach for WDM Optical Multicasting”,

Proc. of IEEE Workshop on High Performance Switching and Routing, May 2001.

[26]. Z. Wang, B. Zhang, C. Lin and C. Chan, "A broadcast and select WDM-PON and its protection.", Optical Communication, 2005. ECOC 2005. 31st European Conference on Volume 3, 25-29 Sept. 2005 Page(s):549 - 550 vol.3

[27]. 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.

[28]. H. Zang, J. P. Jue, Biswanath Mukherjee, "A review of routing and wavelength assignment approaches for wavelength routed optical WDM networks.", Optical Networks Magazine, 1(1):47-- 60, January 2000. 24.

[29]. X. Zhang, J. Y. Wei, and C. Qiao, "Constrained multicast routing in WDM networks with sparse light splitting.", Journal of Lightwave Technology, 18(12):1917.

[30]. http://zone.ni.com/devzone/conceptd.nsf/webmain/

# Vita

Abhishek Satyendranath was born to Dr. Satyendranath and Mrs. Pushpalatha Satyendranath on January 27th 1981 in Bangalore, India. He received his bachelor’s degree in telecommunication engineering from Visveswaraiah Technological University in 2002 after which he worked as a Software Engineer at Infosys Technologies Limited for over a year. He joined LSU in Spring of 2004 to pursue graduate studies in the Department of Electrical and Computer Engineering. He will receive his master’s degree in August 2006.