

## **Acknowledgement**

I would like to thank Dr. Arunita Jaekel, The Natural Sciences and Engineering Research Council (NSERC), and The Computer Research Association's Committee on the Status of Women in Computing Research (CRA-W), to offer this study opportunity to me. I appreciate Dr. Arunita Jaekel's help and guidance. She has been teaching me a lot, not only the fundamental knowledge of the optical networks, but also the research method and scientific attitude.

### **Canadian Distributed Mentor Project (CDMP) Summer 2005 Final report:**

#### **Distributed Dynamic Lightpath Allocation In Survivable WDM Networks**

##### **Abstract**

This project is to investigate a distributed algorithm for solving the dynamic routing and wavelength assignment (RWA) problem using both dedicated and shared path protection techniques for survivable wavelength division multiplexing (WDM) networks. Our goal of this project is to minimize the number of wavelength-links for each new connection, and satisfy communication requests as much as possible based on the given limited network resources. Each network node operates independently using the local information of the network, and communicates to each other by passing control messages. We tested our distributed algorithm by a set of well-known networks. Compared to optimal solutions using ILP formulations, heuristics solutions based on the distributed algorithm are more scalable and faster.

##### **1. Introduction**

Optical networks are attractive candidates for wide-area backbone networks, due to their large bandwidth, low attenuation and low error rates [1]. A lightpath in an optical network is an end-to-end all-optical communication path from a source node to a destination node through a number of intermediate router nodes [5]. Each lightpath must be assigned a route over the physical network, and a specific channel on each fiber it traverses [5]. This is the standard routing and wavelength assignment (RWA) problem [2]. A lightpath can be assigned the same wavelength in all fibers it traverses. This is the well-known wavelength continuity constraint.

There are two strategies to solve the RWA problem: static and dynamic lightpath allocation.

In static allocation, the set of lightpaths to be established is known in advance [2].

In dynamic allocation, lightpath creation is based on the communication requests which arrive at random times. The connection is terminated and the WDM channel is released for future use after a communication is over. Our goal of this project is to minimize the

number of wavelength-links for each new connection, and satisfy communication requests as much as possible, based on the given limited network resources.

The heuristic method and the Integer Linear Programming (ILP) method [3], [4] are two main methods to solve the RWA problem in survivable WDM networks. ILP formulations are not practical for large networks since they need complex computation, massive memory occupation and computational time. However, ILP formulations generate optimal solutions, which can be used as benchmarks to validate and test heuristic methods. This project presents the distributed algorithm to solve dynamic lightpath allocation problem in survivable WDM networks. Compared to optimal solutions using ILP formulations, heuristics solutions based on the distributed algorithm are more scalable and faster.

A network fault, like fiber cut in a single fiber link, will cause serious data loss. That is why the protection schemes are so important in survivable WDM networks. In this project, we use both dedicated and shared path protection techniques. For each new communication request, a primary path and an edge-disjoint backup path are set at the same time. In dedicated path protection, the channels used by a backup lightpath are always reserved for a single primary lightpath[5]. The resources allocated to this path can not be used or shared by any other (backup) lightpath. In shared path protection, also called backup multiplexing, resources may be shared by two or more backup paths if and only if the corresponding primary paths are edge disjoint [5]. So, shared path protection is better than dedicated path protection for optimizing resource utilization.

## **2.Distributed Algorithm**

In the distributed algorithm, each node knows only the information of its direct neighbors, instead of the global information of the entire network. During the process of creating a new lightpath, each corresponding node only communicates with its direct neighbor in the route. So, each node only knows the connections routed through it and the links directly connected to it. In our data structure, there are two main types of information, network information and lightpath information, stored in each node.

### **2.1 Network Information:**

The network information includes the following main fields:

Each node has its own node identifier. It knows the node identifiers of its adjacent nodes, and edge numbers of the outgoing links connected from itself to its adjacent nodes.

Each node  $i$  has a bit-card table to store a set of  $R$  (here  $R=3$ ) link-disjoint alternative routes from node  $i$  to all other nodes in the network. All alternative routes were pre-computed using Dijkstra's shortest path algorithm. While a communication request with source node  $s$  and destination node  $d$  is generated, a specific route is selected for the primary path of the new lightpath, and a link-disjoint route is reserved for the backup path. The method used to choose the primary path and backup path is to choose a route with a largest number of "free" channels on the first edge in the route. The free channel means that the channel has not been assigned to existing lightpaths and is not reserved for a new lightpath either. A link with the largest free channels means less congestion and gives us more successful chances. Although this may not be the best choice on the other edges in the route, it is the reasonable choice under the local situation of the source node.

Each node has a set of outgoing links. Each outgoing link has the same number of channels. A field named CurrentState(an integer number) indicates four different states of a channel on a particular outgoing link.

- 1) CurrentState=0: This channel is available to reserve or assign to a new lightpath.
- 2) CurrentState=1: This channel has been already used by an existing lightpath.
- 3) CurrentState=2: This channel is reserved by another new lightpath passing through this edge. We say the channel is “locked”. If it is not selected for that lightpath, it will be released finally, and the state will be set back to 0 for future use. Otherwise it will be set to 1, indicates this channel is “busy”.
- 4) CurrentState=3: This channel is not only used by one or more existing backup lightpaths, but also locked by the new backup path under the backup multiplexing condition. This state is only used for shared path protection. It will be set back to 1 finally no matter the new backup path is successful or not.

When we search a free channel for the new connection on a particular link, we check the CurrentState of each channel. In dedicated path protection, a channel is considered a free channel if and only if the CurrentState is 0. The CurrentState will be set to 2 if it is locked for the new connection. In shared path protection, except CurrentState 0, if the channel has CurrentState 1 and satisfies the backup multiplexing condition, it is still available for the new backup path. In this case, the CurrentState will be set to 3 when it is locked by the new connection.

Another important field used only for shared path protection is a set of queues of connections on a particular link. We need store the connection queue for each channel on the particular link. The element in the connection queue includes the following information:

- 1) The unique identifier of an existing connection using this channel. It is a combination of the source node identifier and a connection number (a sequence number started from 0).
- 2) A flag (“p” or “b”) indicates that channel is used by a primary path or backup path.
- 3) The route is used as the primary path for this existing connection.
- 4) A pointer pointed to the next element in the queue.

We store the primary route of the new connection in the lightpath information of the source node, and pass the lightpath information by control message to each node in the new selected route. We will discuss the lightpath information and control message later. Based on the above information, it is easy for us to check if the new backup path satisfies the backup multiplexing condition with the existing backup path by checking whether two primary paths are link-disjoint or not.

## 2.2 Lightpath Information:

Each node  $i$  stores all the lightpaths started from itself (as source node  $s$ ) to destination node  $d$  in a connection information queue. Each connection has information about the source-destination node pair, connection number, primary route and backup route used for this connection, the channels used for primary path and backup path, and the state (s indicates successful, f indicates failure, p indicates still in process ) which indicates that a lightpath has already been created, failed or is still in the lightpath-setup phase.

In our distributed algorithm, the lightpath-setup is built by passing the control message, which is also called LP-record, from the source node  $s$  to the destination node  $d$  along the pre-selected route.

### 2.3 Control Message (LP-record):

In our project, we set a probability  $p$  ( $0 < p < 1$ ). At each time slot, a random number is generated. If the random number is less than the predetermined probability, a new communication request is generated. We randomly choose two nodes as source node  $s$  and destination node  $d$  for the new connection ( $s$  is not equal  $d$ ). In the initial connection setup phase, the source node  $s$  selects specific routes with the largest free channels on the first edge from its bit-card table for the primary path and backup path of the new connection. If there are no available routes for both primary path and backup path, we increase the failure number by 1, and claim the new connection has failed. Otherwise, first, we assign a unique identifier using the combination with the source node number  $s$  and the sequence connection number to the new connection. Then we record the lightpath information including source-destination node pair, pre-selected primary route and backup route, and store them in the connection information queue of the source node  $s$ . The lightpath-setup state is set to “p” ( in process ) and the channels for primary path and backup path are set to -1. Finally, we generate two LP-records on the source node  $s$ , one for primary path and another one for backup path. The LP-records will be passed as the control messages with different message types to the next node along the selected route to the destination node  $d$ . A LP-record consists nine main fields:

- 1) source node
- 2) destination node
- 3) connection number
- 4) primary route
- 5) backup route
- 6) message type
  - a) “G”- The LP-record of the new lightpath is just generated on the source node.
  - b) “F”- The LP-record is forwarding and updating in the intermediate nodes.
  - c) “R”- The response message is sent backward to the source node after the channel is selected successfully.
  - d) “r”- The response message is sent backward to the source node when the new lightpath fails. The locked channels on each edge should be released.
  - e) “D”- free resource message. Each new connection needs two paths, one for primary path and another one for backup path. When one path is successfully established and another path fails, the new connection fails. Then, we increase the failure number by 1, and send a free resource message from the source node to the destination node along the successful path to release all resources allocated to the successful path.
- 7) selected wavelength (initialized to  $-1$ , set to selected channel  $k$  at the destination node)
- 8) lightpath type (“p”-primary path, “b”-backup path)
- 9) locked channels: a set of available channels (up to MAXLOCKS) can be locked for a single new connection. MAXLOCKS is a predefined constant within the range from 1 to channel numbers (the number of wavelengths on a single link).

When a node deals with a control message (LP-record), it checks the message type first. Forwarding message (message type “G” or “F”):

If it is a new generated lightpath ( message type “G” ), the source node will generate a random number within the range ( from 0 to channel numbers  $-1$  ) as the start point to search the free channels (up to MAXLOCKS). Choosing a random number instead of 0 as start point can avoid locked channels of different lightpaths too convergent. It increases the successful chance.

If message type is “F”, the intermediate node will check each channel in the locked-channels field of the LP-record on the corresponding edge.

For message types “G” or “F”, if there is at least one available channel on the corresponding edge, we update the locked-channels field in the LP-record. If the next node is not the destination node, we set the message type to “F”, and forward the copy of the updated control message to the next node along the pre-selected route. Otherwise, we set it to “R”.

Response message (message type “r”):

If there is no free channel on the corresponding edge, this new connection fails. We release all the locked channels, inform the previous node by sending a message with type “r”, and delete the LP-record in the current node. Each previous node along the route backward to the source node will release all the locked channels in their locked-channels field and delete the LP-record. Finally, the source node will delete the path information about the failed connection.

Response message (message type “R”):

When the message reach the destination node, the destination node will select an available channel from locked-channels field as the final channel used for the new lightpath. The updated control message will be passed from the destination node to the previous node back to the source node. Each node receiving a response message with type “R” will do the following things:

1. Update the selected-wavelength field of the LP-record.
2. Release all the locked channels except the selected channel in the locked-channels field.
3. Set the CurrentState of the selected channel on the corresponding edge to 1.
4. For shared path protection, we insert the new successful connection information to the connection queue of the selected channel.
5. Pass the response message to the previous node.

We know each new connection has one primary path and one backup path. When the response message of one path is sent back to the source node, the source node will check the state of another path in the lightpath information. There are 3 different situations we need consider:

1. Both paths are successful: The new connection is created successfully. We update the lightpath information in the source node, set the state of the new successful path to “s”, and update the final channel used for the new path.
2. Both paths are failed: The new connection failed. The lightpath information will be deleted from the source node.

3. One path is created, another path failed: The new connection failed. Like situation 2, the lightpath information will be deleted from the source node. The resource allocated to the successful path should be released for future use. A free-resource control message will be sent from the source node to the destination node along the successful lightpath. For dedicated path protection, each node along the successful lightpath sets the CurrentState of the final channel on the corresponding edge to 0. For shared path protection, we delete this path from the connection queue of the final channel on each corresponding edge. If the connection queue is empty, we set the CurrentState of the final channel on that edge to 0. Otherwise, we set the CurrentState to 1.

### 3. Experimental Results:

We tested our distributed algorithm by a set of well-known network using both dedicated and shared path protections. We compared the performance of our distributed algorithm to optimal solutions generated from ILP formulation and heuristic solutions based on the centralized algorithm.

Figure 1 is the physical topology of the National Science Foundation Network (NSFNET). NSFNET has 14 nodes and 21 edges.

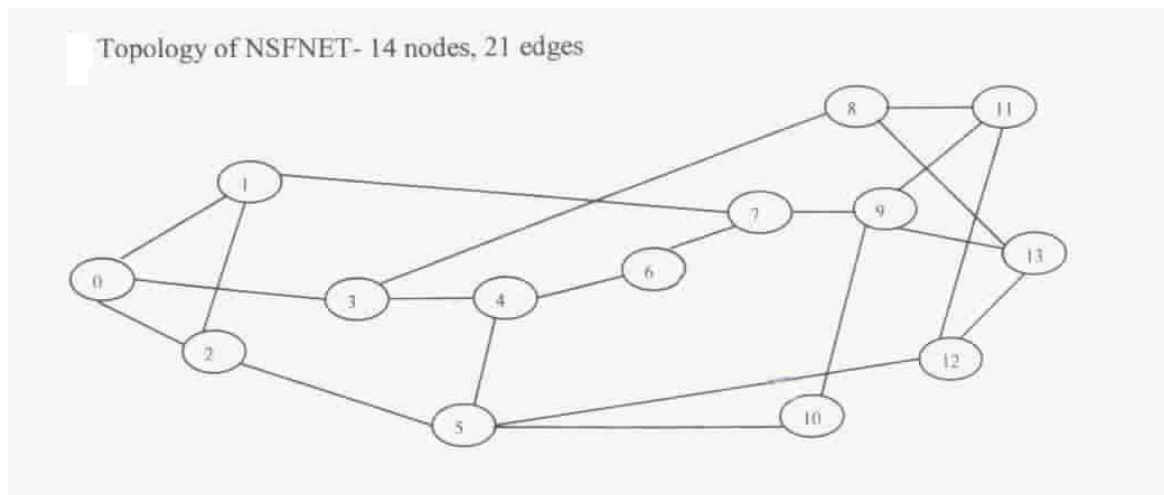


Figure 1 : Topology of NSFNET

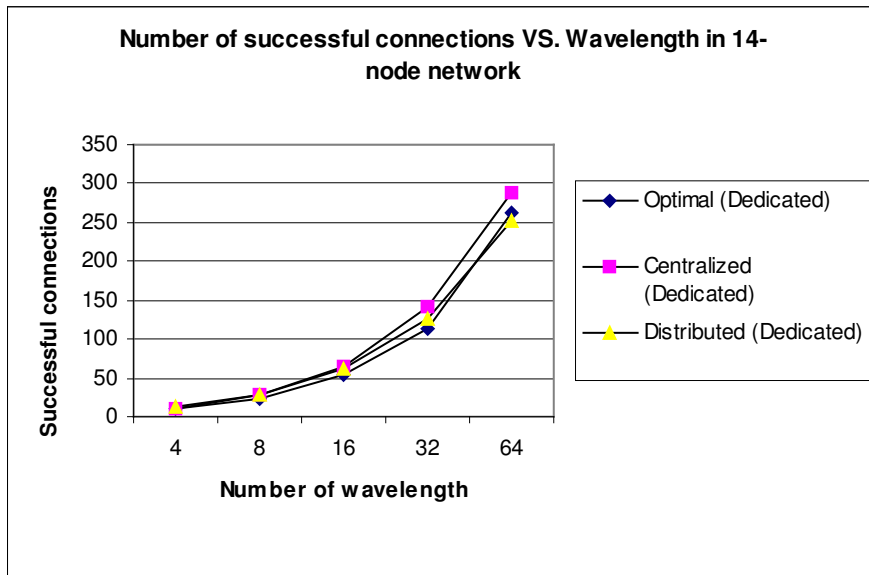


Figure 2 (a): Dedicated path protection

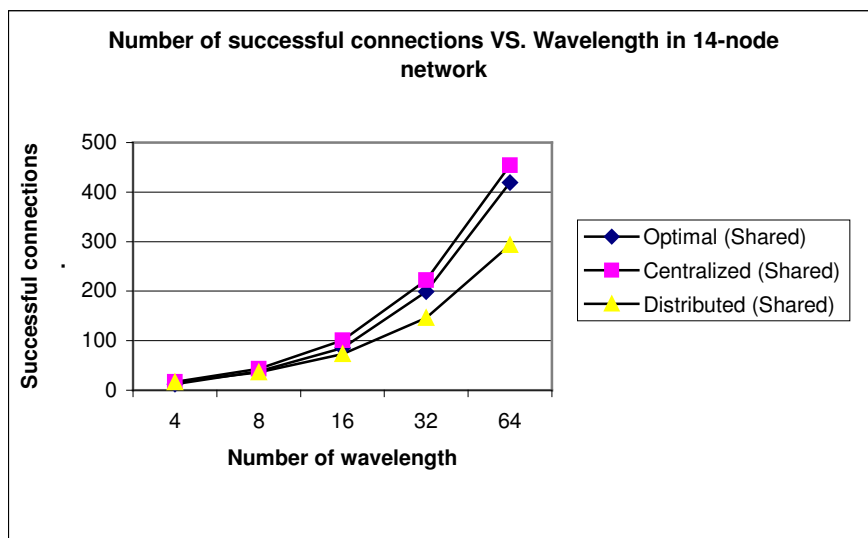


Figure 2 (b): Shared path protection

Figure 2 shows the successful connections we got when the second failure occurred using (a) dedicated path protection and (b) shared path protection. The performance of the centralized algorithm is very close to the optimal solutions, and more connections are blocked based on the distributed algorithm. The reason is that in the centralized algorithm the new connections are setup sequentially, but in the distributed algorithm, several new connections are setup simultaneously. In the distributed algorithm, more free channels are locked by some new connections in the setup phase and can not be used for other new

connections, although finally, they will be released. Another reason is in the centralized algorithm, up to  $R$  ( $R=3$ ) pre-computed routes are reserved for a new connection. If one route fails, we can try other routes. This also increases the successful chance for a new connection. In the distributed algorithm, only one single route is pre-selected for a new lightpath. If it fails, we claim the new connection fails. The main problem with centralized algorithms (both ILPs and heuristics) is that the central agent can quickly become a bottleneck [5].

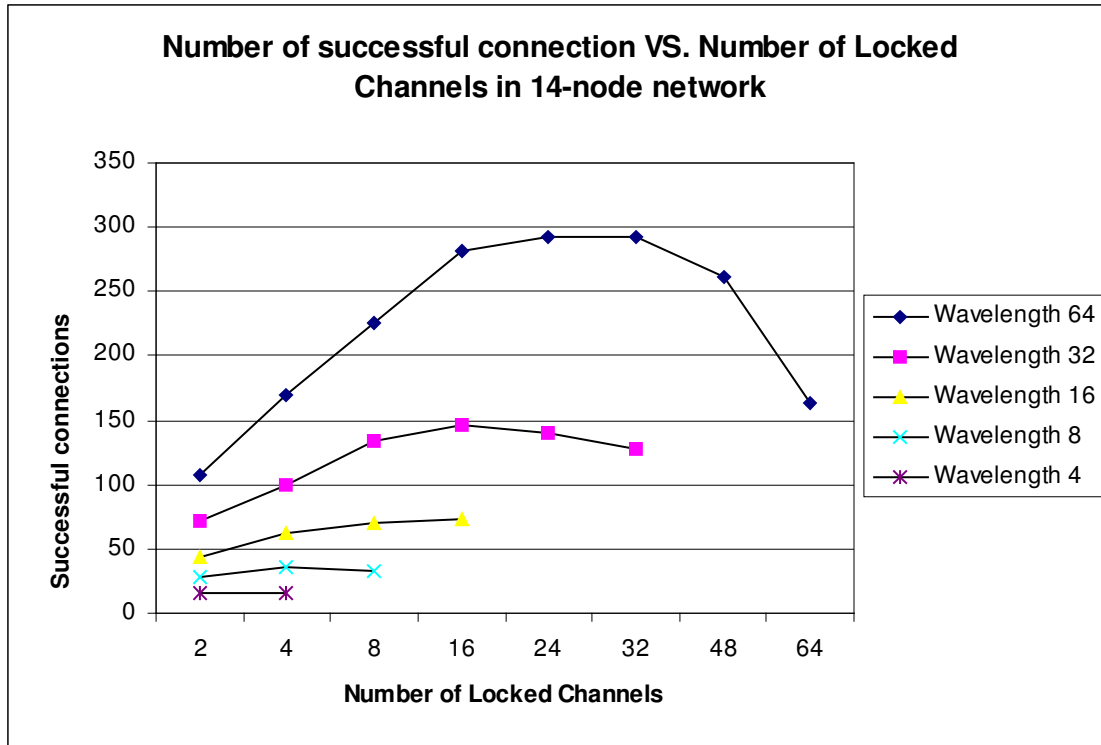


Figure 3: Number of successful connection vs. locked channels in NSFNET

Figure 3 shows the relationship between the successful connections and the number of locked channels (MAXLOCKS) in the distributed algorithm. Too small MAXLOCKS will reduce the successful chance for a given lightpath, but too large MAXLOCKS will block other lightpaths from finding available channels. So it is important for us to choose a suitable MAXLOCKS. We test our algorithm using different MAXLOCKS (from 2 to the maximum number of wavelength per fiber). The results show that we get the best performance when the MAXLOCKS is close to the maximum number of channels.

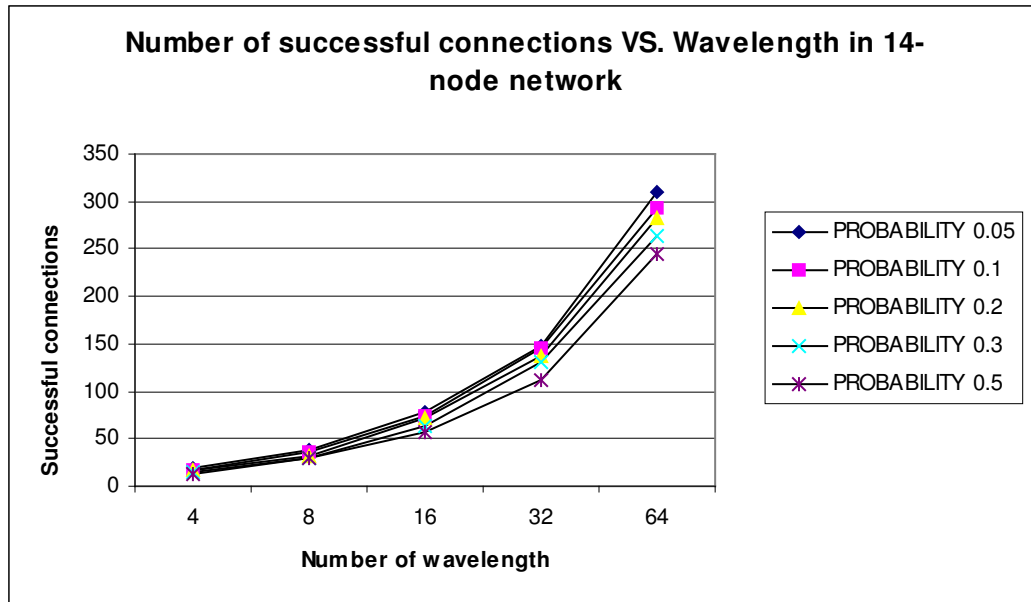


Figure 4: Number of successful connections vs. wavelength in NSFNET

Figure 4 shows the relationship between the successful connections and the pre-set probability in the distributed algorithm. We get better performance as the probability decreases.

#### 4. Conclusions

In this project we use the distributed algorithm to solve the dynamic lightpath allocation problem in survivable WDM networks using both dedicated and shared path protection. Each network node operates independently using the local information and communicates to each other by passing control messages. Compared to optimal solutions using ILP formulations, heuristics solutions based on our distributed algorithm are more scalable and faster. It is a good alternative for practical networks.

#### 5. Future work

In the fall term, 2005, I will continue research on static lightpath allocation in survivable WDM network using both ILP and heuristic algorithm under Dr. Arunita Jaekel's guide.

#### References

- [1] T. Stern, K. Bala, "Multiwavelength optical networks-a layered approach" (Addison-Wesley, 1999).
- [2] H. Zang, J.P. Jue and B. Mukherjee, "A review of routing and wavelength assignment approaches for wavelength-routed optical WDM networks," Optical Networks Magazine, January 2000.
- [3] Y. P. Aneja, S. Bandyopadhyay and A. Jaekel, "An Efficient Protection Scheme for WDM Networks Using Dynamic Lightpath Allocation", in HPC Asia02, Dec. 2002

- [4] S. Zhong and A. Jaekel, "Optimal priority based lightpath allocation for survivable WDM networks," in Int. Conf. on Computers, Communications and Networks (ICCCN04), pp. 17-22, Oct. 2004.
- [5] Dr. A. Jaekel, Y. Chen, "Distributed Dynamic Lightpath Allocation in Survivable WDM Network", 2005.
- [6] Dr. A. Jaekel, " Dynamic Lightpath Allocation in Survivable Multifiber WDM Networks", 2005.
- [7] M. Tornatiore, G. Maier, A. Pattavina, "WDM Network Optimization by ILP Based on Source Formulation", 2002.
- [8] Y. Luo, N. Ansari, "Performance Evaluation of Survivable Multifiber WDM Networks", 2003.
- [9] M. Saad, Z. Luo, "On the Routing and Wavelength Assignment in Multifiber WDM Networks".
- [10] S. Zhong, "Priority Based Dynamic Lightpath Allocation in WDM Networks", May 2004.