Routing and Wavelength Assignment in Optical Networks

Qais M D M Al Rousan

A Thesis

in

The Department

of

Electrical and Computer Engineering

Presented in Partial Fulfillment of the Requirements
for the Degree of Master of Applied Science at
Concordia University
Montreal, Quebec, Canada

June 2004

Library and
Archives Canada

Published Heritage
Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et
Archives Canada

Direction du
Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence
ISBN: 0-612-94690-8
Our file Notre référence
ISBN: 0-612-94690-8

The author has granted a non-exclusive license allowing the Library and Archives Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou aturement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

Canada

Abstract

Routing and Wavelength Assignment in Optical Networks

Qais M D M Al Rousan

Routing and wavelength assignment in optical domain is not an easy task as it is in electrical domain. This fact is due to the optical nature where wavelength labeling is much more than just identification for a specific traffic. It indicates the real physical resources and bandwidth requested by the source. In Generalized Multi Protocol Label Switching (GMPLS), after computation for an appropriate route path, a special attention is required to allocate, instantiate, and swap the appropriate label. Such allocation is done by signaling protocols such as Resource Reservation Protocol with Traffic Engineering (RSVP-TE). Although there have been much work done to allocate and accommodate the requested resources through the optical network, successful labeling assignment could not be always guaranteed unless a reasonable amount of signaling takes place. This is especially if the network core nodes are wavelength conversion incapable. The main contributions of this thesis are, firstly, proposing the Bottleneck Congestion Avoidance algorithm and technique that enhances the use of suggestion label in RSVP-TE to reduce the delay and blocking cost during wavelength assignments; secondly, proposing the Traffic Engineering technique that uses explicit routing and the proposed Proper Suggested Label Selection (PSLS) table to distribute the network resources and to provide fast wavelength allocation. This technique also uses the proposed Lambda Blocking Check Algorithm (LBCA) that inexpensively and efficiently checks resources availability through the optical network. The proposed technique enhances the network utilization by reducing the blocking probabilities.

Acknowledgements

This work would not have been possible without family, doctors and friends supports. It is impossible to list here all those who helped to sustain me during this thesis accomplishment, and I apologize in advance for any omissions.

However, I must give special thanks to my supervisor, Dr. Anjali Agarwal for her technical support and constant encouragements. Her believe in me was the main motivation for me during my research. I would like to thank her for giving me the opportunity to work with her as a research assistant and as a teaching assistant as well.

I would like to thank our Graduate Program Director (M.A.Sc), Dr. M. Reza Soleymani for initially accepting me as a graduate student.

I must admit that if not for the love and every kind of support of my father, Muhie Ddin Al Rousan, my mother, Afaf Al Mardini, my Brother, Med. Dr. Lawrence Al Rousan, my fiancé, Hanady Al Mardini, my aunt Falak Abu Al Enean, I would not be able to complete my M.A.Sc. program at Concordia University. I am particularly grateful for all of them for their love, care and understanding, especially during the busy periods. I consider myself the luckiest person on earth for having my wonderful great parents.

I would like to thank all my truly friends, especially, Qutaibeh Katatbeh, Najem and Wajdi Al Dahabi, for their logical support. Last but not least, I thank Concordia University for giving me the privilege of being one of its graduate students. Everything that I have is given by God, and my eternal appreciation goes to him.

Qais M D M Al Rousan

June 2004

Table of Contents

List of Figures

List of Tables

Chapter 1

1 INTRODUCTION

Since the revolution in computer networking and Internet services, the demand for high speed data traffic and bandwidth became much more imperative, especially to provide some of the most important and popular Internet services such as:

- real-time multimedia traffic.
- video conference.
- massive data capacity downloading and uploading through the Internet in high speed and fabulous time.

The required high speed data traffic and bandwidth can be achieved by using the optical network which allows a tremendous high speed data traffic to be exchanged by means of its intelligent architecture. It is also important to note that one of the most important domains that rely on the optical network and its high data capacity is the medical domain. This is especially to operate some difficult and sensitive operations in the remote regions in real-time from other civil regions where the medical doctors are much more skilled.

1.1 Network Classifications

We can define three network generations depending on their physical layers:

- First generation: Copper-Based or Microwave technologies. In this network, transmission of data and switching are based on electronic domain.

- Second generation: Copper or Microwave links with Optical Fiber technologies. In this network transmission of data is based on optical domain, but switching is still based on electronic domain.
- Third generation: Wavelength Division Multiplexing (WDM) [1, 2, 3, 4] technologies. In this network both transmission of data and switching are based on optical domain.

1.2 Wavelength Division Multiplexing

It is imperative to know that the cornerstone for the optical domain is the use of Wavelength Division Multiplexing WDM. Dense Wavelength Division Multiplexing DWDM [5] is a fiber-optics technique that multiplexes different light wavelengths in one fiber. DWDM creates set of parallel optical channels through single fiber, giving a fabulous capacity of data rate and bandwidth transmission. Some important components in the optical network and the DWDM are:

- Add/ Drop Multiplexer (ADM). It is used to add wavelengths to the system or to drop from it whenever there is a need to do so without any need for the Synchronous Optical Network (SONET) equipment. It could also convert one wavelength or more as a fixed or as a dynamic mechanism.
- Optical Cross Connect (OXC): It provides a switching capability including restoration, monitoring and provision.
- Optical Splitter (OS): is normally used to split the wavelengths which are usually used in multicasting or broadcasting techniques.

1.3 Routing Models in the Optical Network

Two models can describe the optical architecture. First model is an *Overlay* or *network level* model, which consists of edges that hide the internal optical network to provide two kinds of control services. First control service is the control within the optical core. Second control service is the control of the surrounded edge routers. Thus, this model provides an optical cloud within the core. The second model is *Peer* or *link level* model, which describes the internal of the optical cloud. It therefore allows surrounding routers to participate in the routing decisions. Figure 1.1 illustrates both the models.

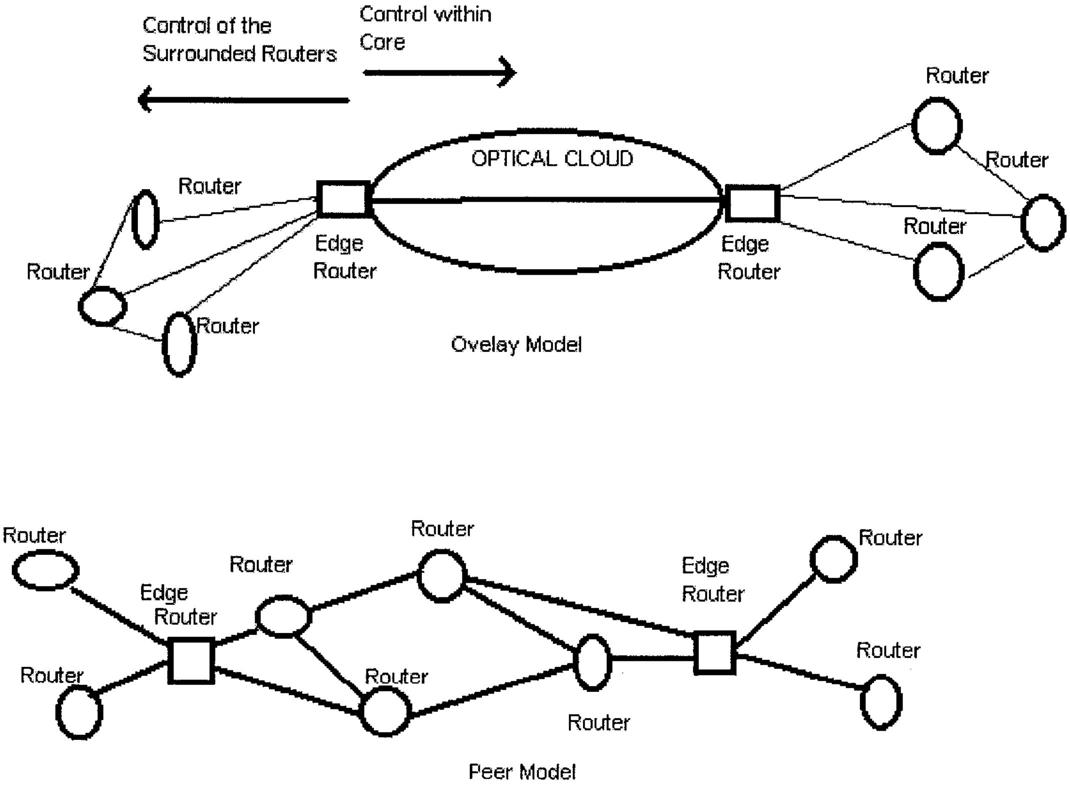


Figure 1.1 Overlay and Peer models

We can define three important characteristics that are needed to be in a common standard control plane as follows:

- Common control plane simplifies operations and management to reduce the operations costs.
- Common control plane provides a wide range of deployment scenarios ranging from overlay to peer.
- Common control plane allows the choice of peer or overlay (or both).

These characteristics are present in the standard protocol GMPLS which is described in the next section.

1.4 Generalized Multi-Protocol Label Switching

The Generalized Multi-Protocol Label Switching (GMPLS) [6] is an intelligent enhancement to Multi-Protocol Label Switching (MPLS) [7] technique that will accommodate MPLS to be used for the optical network. At first, there was a potential use for ATM over DWDM to provide quality of service for the optical network. Furthermore, the use of SONET was also necessary for reliability, bandwidth and multiplexing issues. The use of ATM and SONET may have its advantages; however their services add a lot of redundancy. By the development of GMPLS, we are able to eliminate the use of ATM, SONET and their redundancy by moving their functions to the routers, OXCs and DWDM. As a result, more cost-efficient wide range data streams network may be provided. Figure 1.2 illustrates the evolution toward the photonic, optical, network.

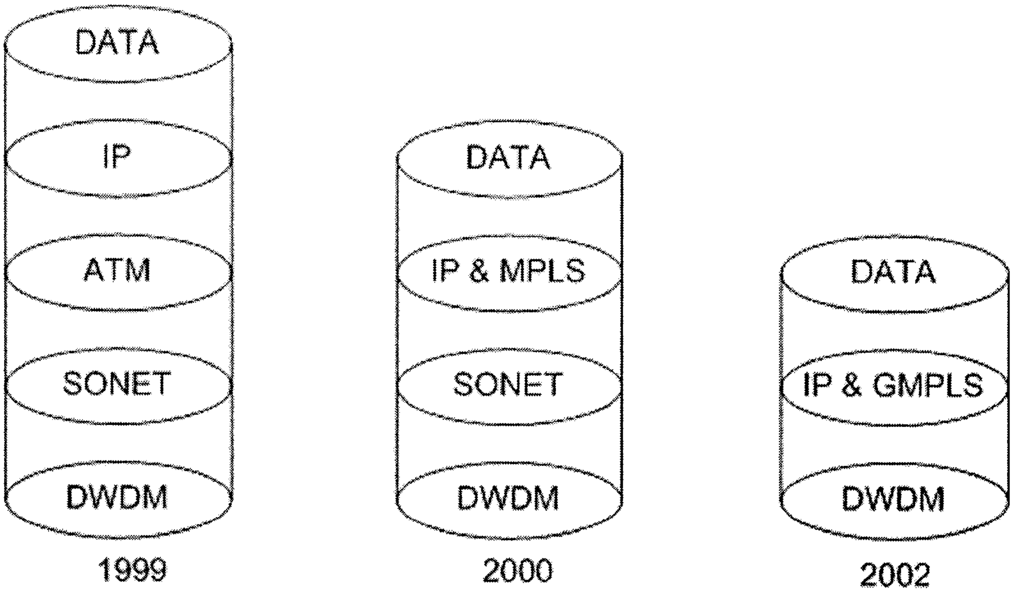


Figure 1.2 **The evolution toward the photonic network**

1.5 Optical Network Motivations and GMPLS Contributions

1.5.1 Routing in GMPLS

To make MPLS compatible for routing use in optical network, GMPLS have made enhancements to MPLS [6, 8, 9, 10]. If the incompatibilities are defined as problems and the enhancements as solutions, consider the following:

MPLS label space is comparatively large (up to million per port), comparing with the limited number of wavelengths, lambdas (hundreds up to thousands). The solution to this problem can be addressed by hierarchical structure for Label Switch Path (LSP). By LSP hierarchy, an LSP can be nested inside other LSPs. Thus, MPLS LSPs that enter the optical network from a node and leave it from the same node type can be aggregated and tunneled using a single lambda. As a result, significant number of LSPs would be conserved. Figure 1.3 illustrates the clouds as a result of hierarchical structure.

Bandwidth is also a great concern. When an optical LSP is setup, it gets a discrete bandwidth (e.g. 2.5 Gb/s) as a channel in fiber. However, it is considered not efficient to leave the entire bandwidth for only one MPLS LSP. The LSP hierarchy is also the solution for this problem. By treating the lambda as a channel in the fiber, the bandwidth can be allocated for a bunch of MPLS LSPs.

Each node in the network has an identical link state database containing information about the LSPs and the nodes surrounded. This indicates that the optical network database can easily be several orders larger than the MPLS network database. The solution to this is addressed by Link Bundling which is an aggregation for several parallel links (fibers) of similar characteristics and assigning them to a single bundled link, thus decreasing the size of the database to a significant size.

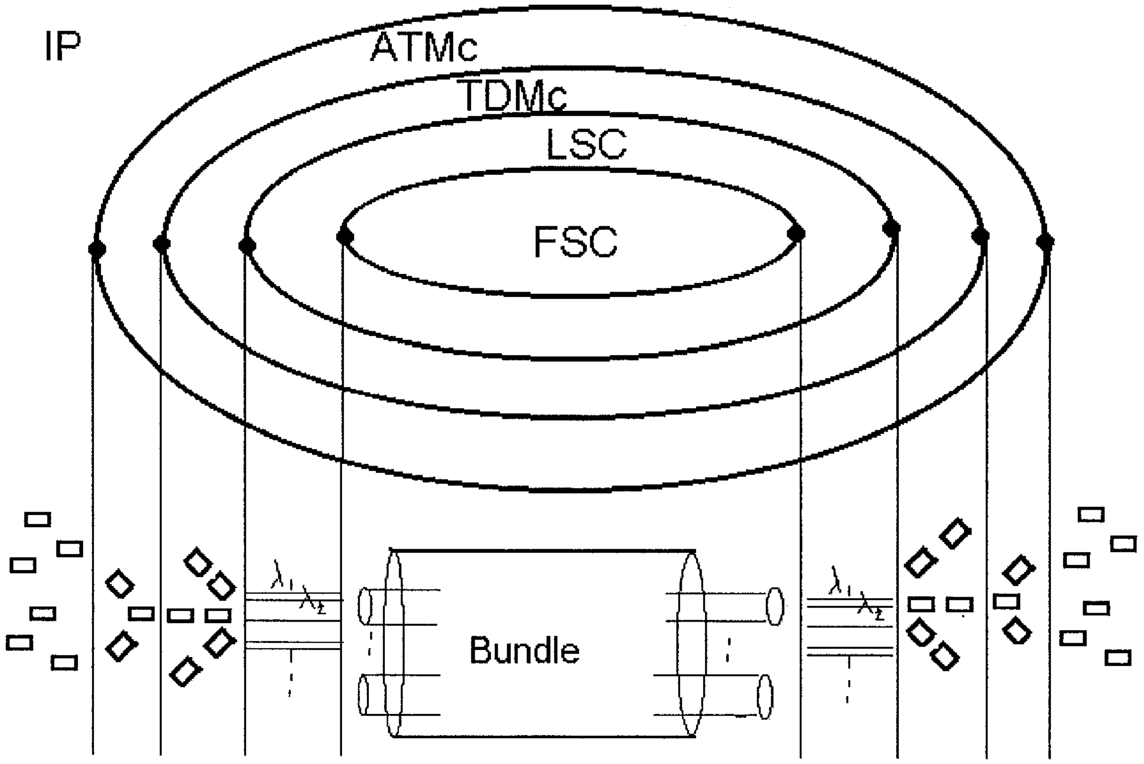


Figure 1.3 Hierarchy map for optical network

Each link in MPLS has its own IP address. Assigning a distinct IP address for each link in optical network is a very big concern because of the huge number of optical links and the limitation of IP addresses. Furthermore, management adds another serious concern. Unnumbered Links is the solution for this problem that dedicates each internet node to number its links locally, where each node has its unique ID. By this mean, links are identified easily by the router ID with the link number.

As a result of the big number of fibers that connect each coupled nodes together, there is a serious concern to identify the connections between the ports. The solution to this is provided by Link Management Protocol (LMP) [11], which runs between adjacent nodes providing the following functions: control channel management, link connectivity verification, fault isolation and link property correlation (e.g., link IDs, priorities and protection mechanism). The use of LMP eliminates the manual need to configure associations between adjacent nodes.

In optical network the data and the control are physically isolated in order to increase the efficiency of the network. LMP addresses this problem by decoupling the control channel from the data channel. Each of them will be transmitted by separate channel (differentiate lambda and/or fiber within the same bundle or even different bundles). An electrical cloud could also be used for signaling transmissions. As a result, there will not be a fault correlation between the data and the control plane.

To keep up with the potential data rate provided by optical networks, fast fault detection and fast switching to alternate channels are needed. LMP has its mechanism to isolate links and channel failures and provide protection by alternate channels and links in a fast way.

1.5.2 Signaling in GMPLS

Enhancements to MPLS signaling mechanisms [12] are needed to add other features to signaling protocols such as Resource Reservation Protocol (RSVP) [13, 14, 15, 16] or Constraint-based Routed Label Distribution Protocol (CR-LDP) [17, 18], to setup the connection, maintain it and terminate it. For this purpose, four basic features have been considered in signaling control plane: to deploy both Overlay and Peer models; to reduce operational costs by simplifying the network operations and management; to use the present routing and signaling protocols; and to satisfy the routing enhancements that have been done by GMPLS, and adding other features to MPLS that can handle the potential data rate in optical networks.

Figure 1.3 is extended in Figure 1.4 to map the reservation and labeling process to setup connection between two terminals using RSVP signaling, where the LSPs have to get through the four clouds as illustrated in the figure. Dismissing the IP overall cloud, R0.2-R0.1, R1.2-R1.1, R2.2-R2.1, and R3.2-R3.1 are the edge routers for the ATM cloud, TDM cloud, LAMBDA cloud and the photonic cloud respectively. Terminal B (TB) requests a connection by sending Request1 (Req1) message from R0.2 toward R1.2. This request message has information about the requested resources and implicates the generalized label requested. Moreover, as an enhancement to MPLS, GMPLS uses suggestion label in the path message. The suggested label allows the upstream nodes to suggest label configuration to the downstream nodes thereby reducing the setup latency especially in the optical cloud (lambda and photonic). However, the downstream still has the right to accept or reject the suggested label given by the upstream. In the reservation process, if the downstream accepts the suggested label, the reservation procedure considers it as acknowledgment and confirms to the suggested label. If not, the downstream sends a reservation message using a new label to the upstream and the upstream has to accept it. Otherwise, the reservation is rejected and a reservation error is sent to start the deletion for the path and resources that have been established and reserved so far.

In this example, it is assumed that the downstream nodes accept the suggested labels that have been suggested by the upstream nodes. When Req1 message with the suggested label switch path1 (SLSP1) is sent through R0.2 toward R1.2, it triggers Request2 (Req2) message with its suggested label (SLSP2) and Tunneled Req1 message (Req1 (T)). The same procedure, triggering and tunneling, will be for Request3 (Req3) message and Request4 (Req4) message as shown in Figure 1.4. However, when Req4 message with its

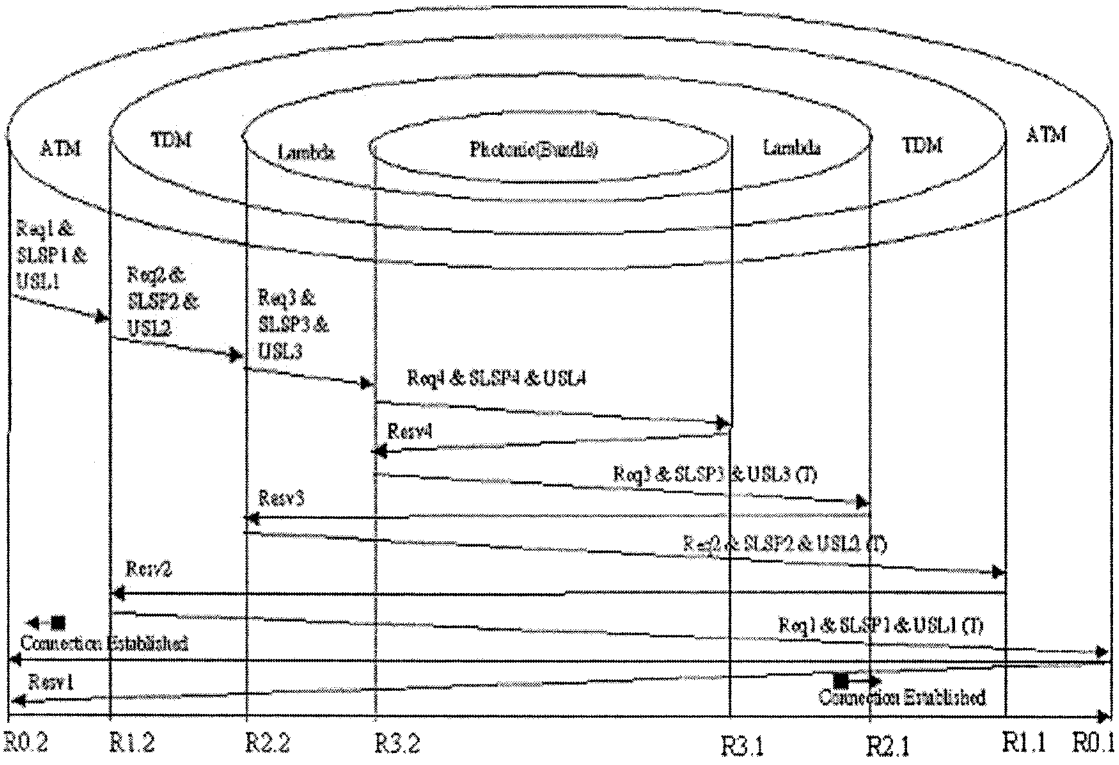


Figure 1.4 Signaling in GMPLS with RSVP-TE

suggested label reaches R3.1, R3.1 will map the Reservation (Resv4 (LSP4)) between R3.1 and R3.2. When the reservation message reaches R3.2, it triggers the Tunneled Req3 message (Req3 (T)) toward R2.1, requesting the resources from R3.1 to R2.1. R2.1 will continue reservations by Resv3 message that will reserve the path from R2.1 to R3.1 and is transparent through the path from R3.1 to R3.2 (already reserved by Resv4 message), and then continue its reservations from R3.2 to R2.2 (ending by that the reservation for LSP3). The same procedure (triggering and reservation) continues until the establishment for LSP2 and LSP1 completing the reservation of the LSP1 successfully, and establishing the logical connection from terminal B to terminal A.

1.5.2.1 Bidirectional LSP Setup

GMPLS adds another significant feature of bidirectional LSP setup to RSVP signaling protocol [19]. In this case, the upstream node sends another label called Up-Stream Label (USL) in the path message to the downstream node. USL must be accepted by the downstream node to reserve and establish the connection on the other direction of the path message establishing a bidirectional connection. However if the downstream node could not accept the upstream label for any reason (such as unavailability of resources or label requested), a negotiation between the downstream and the upstream must take place to try to accommodate another upstream label which is acceptable by both of them. If such a negotiation fails, the downstream will reject the label and sends a reservation error to start the deletion for the path and resources that have been established and reserved so far. Note that USL is also tunneled in path message and triggered as in SLSP. See Figure 1.4 for illustration. There are many advantages to bidirectional LSP such as decreasing the setup establishment latency for LSPs, lowering the overhead requirements, simplifying the selection process for the route and making the connection much more manageable.

1.6 Thesis Motivation and Objectives

Optical domain is one of the most important areas in telecommunications. Its importance comes from its capability and capacity of providing an ultra high speed of information sharing. The information and data are transferred using wavelengths (labels) through the optical fibers. Routing and wavelength labeling assignment in optical domain is not an easy task as it is in electrical domain. This fact is due to the optical nature where label means much more than just identification for a special traffic. It goes further beyond that where it indicates the real physical resource and bandwidth requested by the source. Furthermore, some or all of the optical core nodes could be conversion incapable. When a core node receives a reservation request message in its incoming port from a downstream node that is requesting a specific wavelength, the core node will try to reserve the same wavelength in its outgoing port toward the next node in the route. In the case where the core node is conversion capable, if the core node can not find the requested wavelength available in its outgoing port, it will accommodate the incoming and outgoing wavelengths by converting the incoming wavelength to other available wavelength in its outgoing port. On the other hand, in the case where the core node is conversion incapable, if the core node can not find the requested wavelength available in its outgoing port, it will fail the reservation request. As a result, the connection request will be blocked.

GMPLS is the protocol that governs every aspect of an optical connection such as connection establishment, path reservation, connection tears, data transfer, and fault handling. By the development of GMPLS, we are able to eliminate the use of ATM, SONET and their redundancy. As a result, more cost-efficient wide range data streams network can be provided. In GMPLS, after computation for an appropriate route path, a special attention is required to allocate, instantiate, and swap the appropriate label. Such allocation is done by signaling protocols such as RSVP-Traffic Engineering (TE) [19, 20, 21]. Although there have been much work done to allocate and accommodate the requested resources through the optical network that aim to reduce the blocking probability, for example by accommodating the use of RSVP-TE to optical domain through the use of suggestion label, successful labeling assignment could not be always guaranteed unless a costly reasonable amount of signaling takes place. This is especially if the network core nodes are wavelength conversion incapable. As a result, cost efficient routing and wavelength assignment techniques to reduce the blocking probability cost are challenging and required.

The research work in this thesis aims to reduce the blocking probability cost in the optical domain by using cost efficient routing and wavelength assignment techniques with respect to GMPLS and the latest technologies in the optical domain.

1.7 Thesis Organization

In Chapter 2, we go through the latest works and enhancements that have been done in the optical network to enhance the utilization and to reduce the blocking probability. In Chapter 3, we propose the “Bottleneck Congestion Avoidance in GMPLS Networks” algorithm to enhance the use of suggestion labels in RSVP-TE, therefore enhancing the resource utilization in the optical network. In Chapter 4, we propose the “Routing and Wavelength Assignment in IP Over WDM”. It is a traffic engineering technique for the optical network where the core nodes are wavelength conversion incapable. This traffic engineering technique reduces the blocking probability and enhances the optical network utilization easily and efficiently through its algorithms. In Chapter 5, we build an optical network model with respect to software engineering and object oriented point of view to illustrate the efficiency of the proposed traffic engineering technique by simulation results. At last, in Chapter 6, we conclude our research works and results and introduce our future works prospective.

Chapter 2

2 LITERATURE REVIEW

In electrical domain, process of labeling assignment does not indicate the resources needed to be reserved. However, in optical domain this process indicates the exactly needed and requested resources for the connection request that sent through the path message. Furthermore, it indicates the physical environment and links being requested. This could cause a rejection for the requested label especially if the node is a wavelength conversion incapable, even though resources are available at the node from where label might be requested. In this Chapter, we go through the latest works and enhancements that have been done in the optical network domain to enhance the utilization and to reduce the blocking probability. Several solutions are presented in Sections 2.1 to 2.4; each solution has its strengths and weaknesses.

2.1 The Suggested Label in RSVP-TE Signaling of GMPLS

The proposed solution in [19] and as described in [22], suggested label to enhance the RSVP signaling in GMPLS by allowing the upstream nodes to suggest label configuration to the downstream nodes thereby reducing the setup latency and the blocking probability in the optical network.

Consider the optical network portion shown in Figure 2.1 which assumes four wavelengths available in each link: Lambda1, Lambda2, Lambda3 and Lambda4. The number of links between each pair in the network is one. *A5* and *B* are ingress nodes, *F* and *I* are egress nodes, and *A4*, *A3*, *A2*, *A1*, *A*, *C*, *D*, *E*, *G*, *H*, are intermediate nodes where all the nodes are wavelength conversion incapable. Assume a connection between *A5* and *F* is already established through *A4*, *A3*, *A2*, *A1*, *A*, *C*, *D*, *E* using Lambda1. If *B* requests a connection to be established with *I* through *D*, *E*, *G*, *H* using suggested Lambda1, the suggested Lambda1 will be denied by *D*. In this case, *D* will suggest other available lambda (for example, Lambda2, if the label set in path message from *B* to *D* allows such suggestion) in its path message to *E*. Path message will continue from *E* to *G*, *H*, and *I*. Reservation will proceed from *I* to *B* through *H*, *G*, *E*, and *D* for Lambda2.

As it is clearly shown in the above example, the use of the label set with the suggested label reduces the blocking probability. However, it does not completely eliminate it or guarantee successful labeling assignment. For example, there is no guarantee that reservation for Lambda2 from *I* to *B* will succeed. If *A5* requests Lambda2 for a new connection between itself and *F*, and was able to reserve the link between *E* and *D* successfully before a reservation acknowledgment is received by *E* from *G* (for the first request made to establish a path from *B* to *I* through *D*, *E*, *G*, *H*), the reservation process for the first request will fail. On the other hand, reservation from *F* to *A5* might fail on node *E*, to reserve the link between *E* and *D*, if reservation has already succeeded for the other request from *I* to *B* (while *F* to *A5* reservation is still on progress).

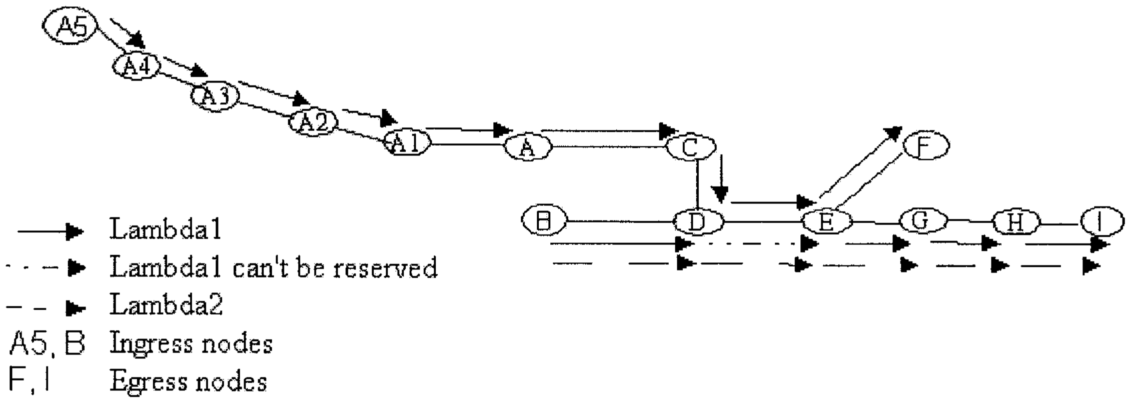


Figure 2.1 Wavelength assignments in optical network

We conclude that although the use of the suggested label reduces the blocking probability, successful labeling assignment could not be always guaranteed.

2.2 Centralized T-E Server with Routing Decision

In this approach, network nodes frequently update Traffic Engineering (T-E) server (through signaling channels) with their resources states (wavelengths availability). Depending on the traffic volume and available resources in each node, T-E server makes the best use of the network resources by reconfiguring the topology of its light paths with changes in the traffic pattern [23]. The advantage of this approach is the decreasing of the blocking probability due to the fact that T-E server would be able to avoid congested nodes. However, frequent traffic updating inefficiently utilizes the network resources. Moreover, congestion due to frequent updating or impaired dynamic provisioning due to traffic updating delay adds as another disadvantage.

Figure 2.2 illustrates a possible reconfiguration (rerouting) by the T-E server for an already established connection due to a possible congestion occurrence in a node through the already selected route.

2.3 Asynchronous Criticality Avoidance Protocol

Using the Asynchronous Criticality Avoidance Protocol [24], core nodes update ingress and egress nodes through signaling channels with their wavelength link states whenever they inquire by probing mechanism. Whenever occupancy of wavelength channels causes a bottleneck between source and destination pair with a width narrower

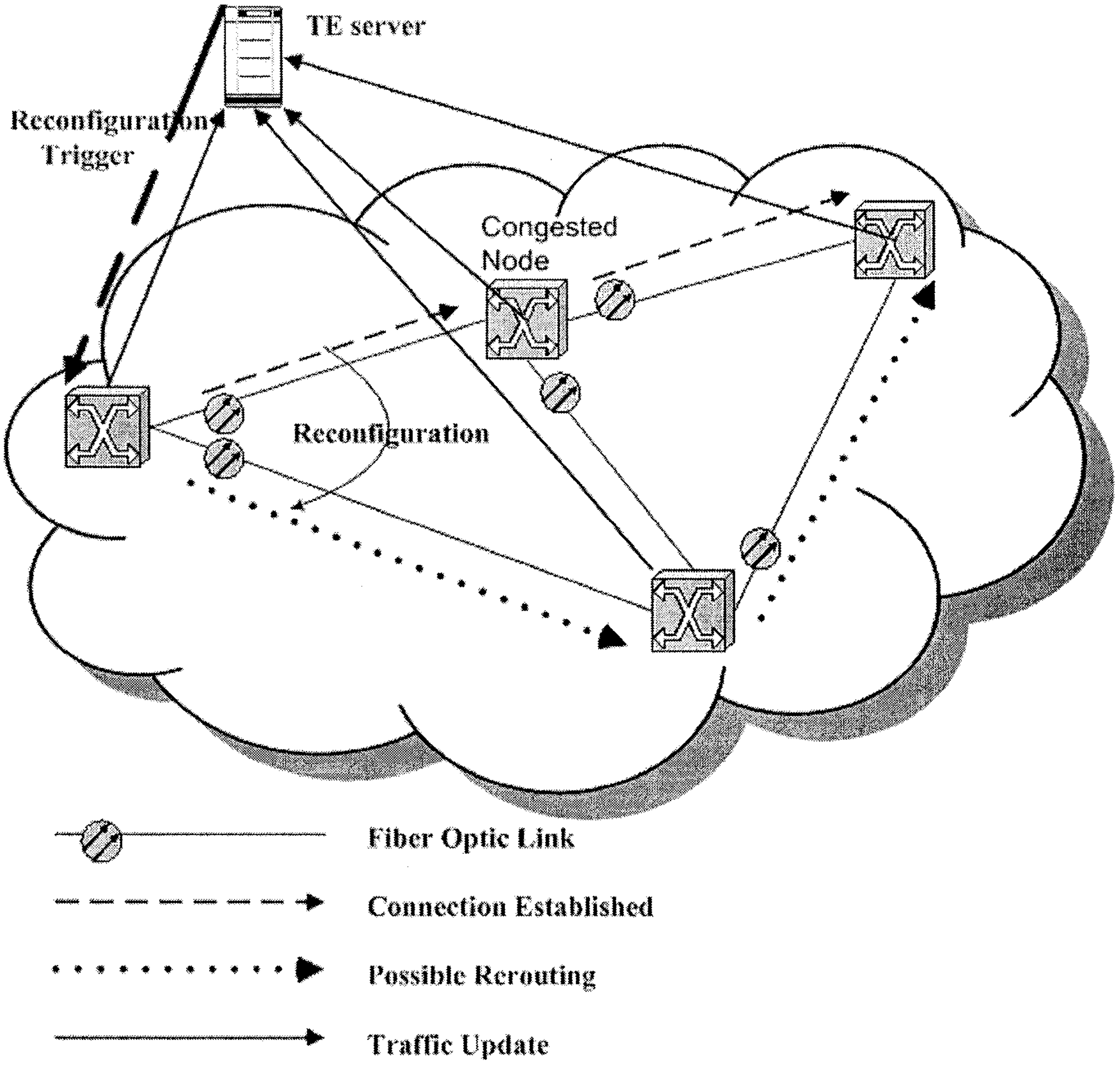
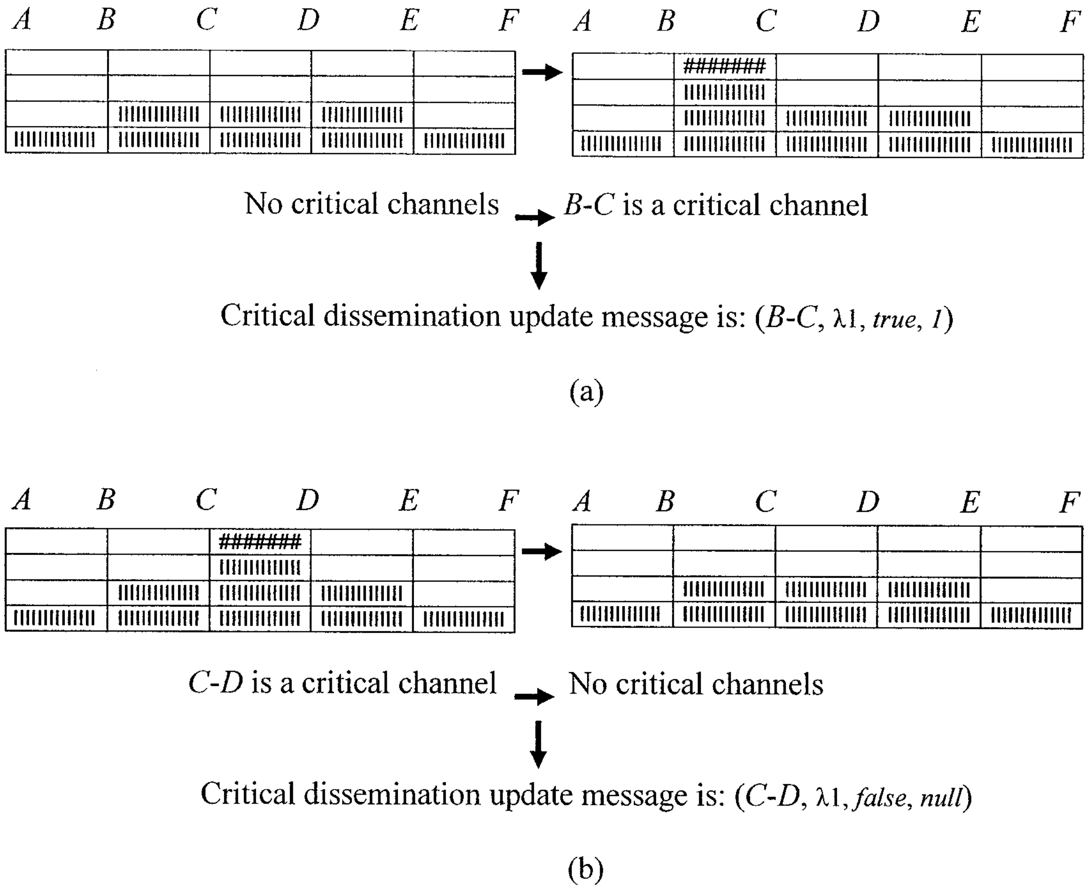


Figure 2.2 **Reconfiguration due to a congestion possibility**

than predefined threshold value, the source and destination mark the bottleneck channel of the link between them as a critical and update the marking decision to the rest edge nodes. By marking a channel as a critical for a source and destination pair on a wavelength, other edges will try to avoid using it in their future routing decisions. By this way, links can be reserved to be occupied by the particular source and destination pair to reduce the blocking probability between them.



No critical channels → *B-C* is a critical channel

Critical dissemination update message is: (*B-C*, λ1, *true*, *1*)

(a)

C-D is a critical channel → No critical channels

Critical dissemination update message is: (*C-D*, λ1, *false*, *null*)

(b)

▥ : Channel occupied by λ1 in a fiber.
☐ : Channel is not occupied by λ1 in a fiber.
####### : Channel is marked as critical.

Number of available fibers between node pairs from source *A* to destination *F*: four.
Number of available wavelengths in each fiber is one: λ1.
Predefined threshold: one.

Figure 2.3 Example of Asynchronous Criticality Avoidance Protocol

Consider the Asynchronous Criticality Avoidance Protocol example in Figure 2.3. Assume the number of the available fibers in the link between the source *A* and the destination *F* is four. The number of the available wavelengths in each fiber is one (λ1). The predefined threshold is one. In Figure 2.3(a), the critical dissemination update message (*B-C*, λ1, *true*, *1*) is released when the residual working capacity (the number of light paths that can be set up between source *A* and destination *F* at a time) becomes equal to the predefined threshold. Here *B-C* indicates the critical channel, λ1 indicates the wavelength concerned in the critical channel, *true* indicates that the defend critical channel in this update message is a new critical channel, *1* indicates the predefined threshold.

In Figure 2.3(b), the critical dissemination update message (*C-D*, λ1, *false*, *null*) is released when the critical channel *C-D* is no more considered as a critical channel because the residual working capacity between the source *A* and the destination *F* becomes bigger than the predefined threshold. Here *C-D* indicates the critical channel, λ1 indicates the wavelength concerned in the critical channel, *false* indicates the cancellation for the defend critical channel.

The advantage of this approach is blocking reduction due to the fact that ingresses are able to avoid possibility of choosing inappropriate suggested labels by marking (reserving) the links to the benefit of the appropriate ingresses. However, this approach has its disadvantages such as the impaired dynamic provisioning due to update delay. Furthermore, simultaneous selection of a suggested label by two ingresses or more may cause conflict in labeling selection. Moreover, the probing mechanism and the wavelength link states update messages are inefficiently wasting the network resources. Also, simulations have shown that this protocol becomes less effective when traffic load increases [24].

2.4 Centralized T-E Server with QoS and Explicit Routing

The approach in [25] proposes a mechanism to support multiple services transport in lambda labeling capable networks. First, the virtual topology is made by routing the paths over the physical topology. Then, the wavelengths are assigned dynamically to the light paths for multiple service classes. An ingress determines the virtual path that a new flow needs to be routed through depending on its Quality of Service (QoS) requirements. The objective of this approach is to find an appropriate route to the requested flow by satisfying the requested QoS and avoiding the links that suffer from maximum congestion. To determine the explicit route, a Traffic Engineering (T-E) server is needed and it has to be updated with the current topology and the available wavelengths in each node of the network. It uses this information to compute the explicit routes in response to the ingress request. It then provides the ingress node with its computation results and the ingress node takes its local decision.

However, updating the centralized routing server must be repeated in a daily or hourly basis depending on the administrator configuration. The advantage of this approach is less blocking compared to the Open Short Path First (OSPF) [26, 27]. Disadvantages are congestion if updating is required to be repeated more often, simultaneous selection of a suggested label by two ingresses or more could cause a label selection conflict, and impaired dynamic provisioning concern depends on the periodically update period.

2.5 Summary

This Chapter introduced several proposed solutions to reduce the blocking probability in the optical network by increasing the routing and wavelength assignment efficiency. It highlighted the advantages and the disadvantages of the proposed solutions. It also provided major challenges and motivations.

In the next Chapter, we will introduce our contribution to reduce the blocking probability and to enhance the resource utilization in the optical network by proposing the "Bottleneck Congestion Avoidance in GMPLS Networks" algorithm to enhance the use of suggestion labels in RSVP-TE.

Chapter 3

3 THE BOTTLENECK CONGESTION AVOIDANCE IN GMPLS NETWORKS

As it is mentioned in Chapter 2, in optical domain labeling assignments (described in Section 1.5.2) indicate exactly the needed and requested resources for the connection request send through path message. Furthermore, it indicates the physical environment and links being requested. This label's indication could cause a rejection for the requested label especially if the node is incapable of wavelength conversions, even though resources are available at the node from where label might be requested. Although the use for the label set with the suggested label in RSVP-TE reduces the blocking probability, we provided in Section 2.1 its limitation to guarantee successful labeling assignment.

In this Chapter we propose a Bottleneck Congestion Avoidance technique to enhance the use of suggested label in RSVP-TE signaling in GMPLS networks when the network core nodes are wavelength conversion incapable.

3.1 The Bottleneck Congestion Avoidance

The Bottleneck Congestion Avoidance technique aims to enhance the use of suggested label in RSVP-TE signaling by improving its limitation that we discussed in Section 2.1 considering the need for the reduction of blocking and connection establishment latency.

Assume two requests with some priorities, Req1 and Req2 are received by an intermediate node where Req1 arrived before Req2 and both requests suggest the same wavelength and request the same output (next node). Assume also that the intermediate nodes are incapable of wavelength conversions and have one link (fiber) left to the next requested node by Req1 and Req2. To solve the assignment conflict, priority must be given to the first request (Req1). However, the intermediate node must try to accommodate and suggest other label (if resources are applicable and label set allows such accommodation) in its output path message for the second request (Req2). This solution would reduce the blocking risk for both requests. However, the second request might experience a reasonable delay due to the need of readjustments for the previous successfully adjusted nodes (to the suggested label) through the already traversed path. Moreover, if Req2 has no more wavelengths in its label set further than the one it suggested, this will cause a blocking for Req2 rather than leaving a chance for the request to proceed and successfully continue reservation (in case that Req1 did not succeed in its reservation process, or even if reservation for Req2 was faster than for Req1). Furthermore, a taken decision does not take in consideration the efforts (and time) that have been spent so far in adjusting the routers mirrors for successfully suggested labels.

In order to solve such matter when no wavelengths are available in the label set of Req2 to be accommodated, our proposal specify that decision must be taken and priority must be given to the request that more likely have had successfully traversed more nodes. The successfully traversed node is defined as follows.

Successfully Traversed Nodes for a Request Just received (STNRJ), Req2 in our example, is defined as the number of successfully (suggested label) traversed nodes in a row (without interruption) from a node through the request path till the intermediate conflict node that has to take the decision. Successfully Traversed Nodes for a Request

Already received (STNRA) but its reservation not yet acknowledged, Req1 in our example, is defined as the number of successfully traversed nodes in a row (without interruption) from a node through the request path till the intermediate conflict node plus the rest number of nodes to be traversed after the conflict node along path message. Assume:

N := the total number of nodes of a path (route) from source (ingress) to destination (egress)

F := the number of nodes from source (ingress) to conflict node (included) along the route

S := the number of successfully traversed nodes in a row from intermediate node till the conflict node (included).

The path message is extended by adding an F and S fields. This F field has an initial value equal to one. Every intermediate node increments the received value of F by one. S field also has an initial value of one. Every intermediate node updates the S field value as following:

S := S+1, if the suggested label requested by the previous node is acceptable by the intermediate received node (i.e. if it is able to match the incoming wavelength (channel) with the outgoing wavelength).

S := 1, if the suggested label requested by the previous node is unacceptable by the intermediate received node (i.e., if it is unable to match the incoming wavelength (channel) with the outgoing wavelength).

In conversion incapable network, if an intermediate node receives two requests, Request1 (Req1) and Request2 (Req2), for the same lambda and for the same last available output link where Req1 arrived first, the intermediate node must take its decision with respect to the Bottleneck Congestion Avoidance algorithm in GMPLS Networks as illustrated in Figure 3.1 and its pseudo code in Figure 3.2.

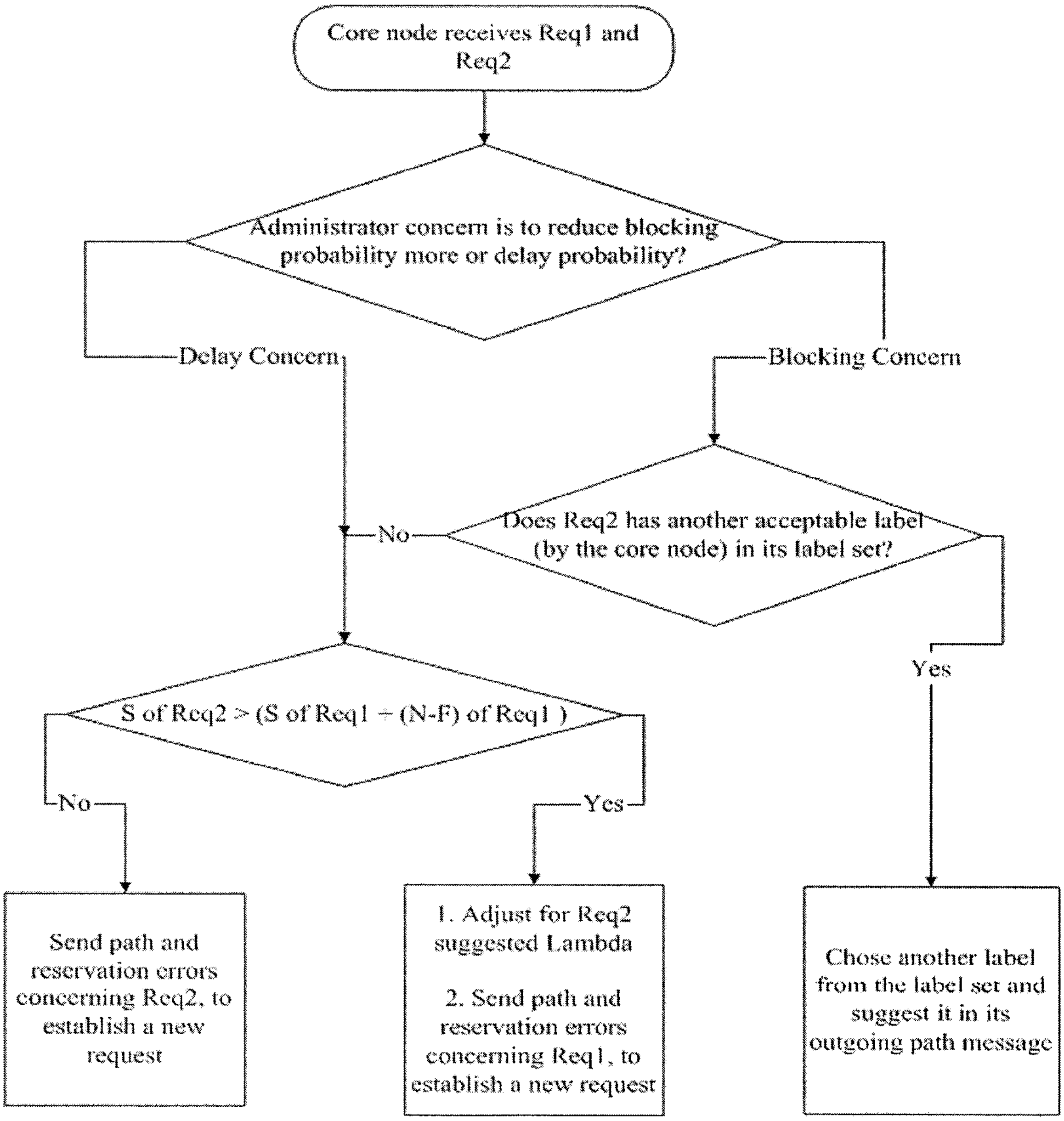


Figure 3.1 The Bottleneck Congestion Avoidance algorithm

```
If (Administrator concern is to reduce blocking probability rather than delay
   probability)
{
   If (Req2 has other acceptable label (by the node) in its label set)
   {
      choose other label from the label set and suggest it in its outgoing path message;
   }
   else
   {
      If (S of Req2 > (S of Req1 + (N-F) of Req1)
      {
         Adjust for Req2 suggested lambda;
         Send path and reservation errors concerning Req1, to establish a new request;
      }
      else
      {
         Send path and reservation errors concerning Req2, to establish a new request.
      }
   }
}
else (Administrator concern is to reduce delay probability rather than blocking
     probability)
{
   If (S of Req2 > (S of Req1 + (N-F) of Req1)
   {
      Adjust for Req2 suggested lambda;
      Send path and reservation errors concerning Req1, to establish a new request;
   }
   else
   {
      Send path and reservation errors concerning Req2, to establish a new
      request;
   }
}
```

Figure 3.2 The Bottleneck Congestion Avoidance pseudo code

For example, consider the Bottleneck Congestion Avoidance optical network example illustrated in Figure 3.3. Assume that ingress *A* requested a connection (ConReqA) toward egress *D* through intermediate nodes *B* and *C;* ingress *E* requested a connection (ConReqE) toward egress *N* through intermediate nodes *F*, *G*, *H*, *I*, *J*, *K*, *B*, *C*. *L* and *M*.

Assume that ingress *A* requested λ1 for ConReqA in its bath message and the intermediate node *B* was able to accommodate λ1 in its path message for ConReqA toward node *C*; ingress *E* requested λ2 for ConReqE in its path message but the intermediate node *F* was unable to accommodate λ2 for ConReqE in its path message toward *G* and suggested λ1 instead. Assume that λ1 will successfully be accommodated in the ConReqE path messages from *F* to *B*. Assume that at the time when *B* receives ConReqE, it has only one λ1 available between itself and *C* and it does not receive the reservation message concerning ConReqA yet. In such case, the intermediate node *B* (conflict node) has to take a decision regarding the both connection requests (ConReqA and ConReqB) using the Bottleneck Congestion Avoidance algorithm in GMPLS Networks as illustrated in Figure 3.1 and its pseudo code in Figure 3.2 where:

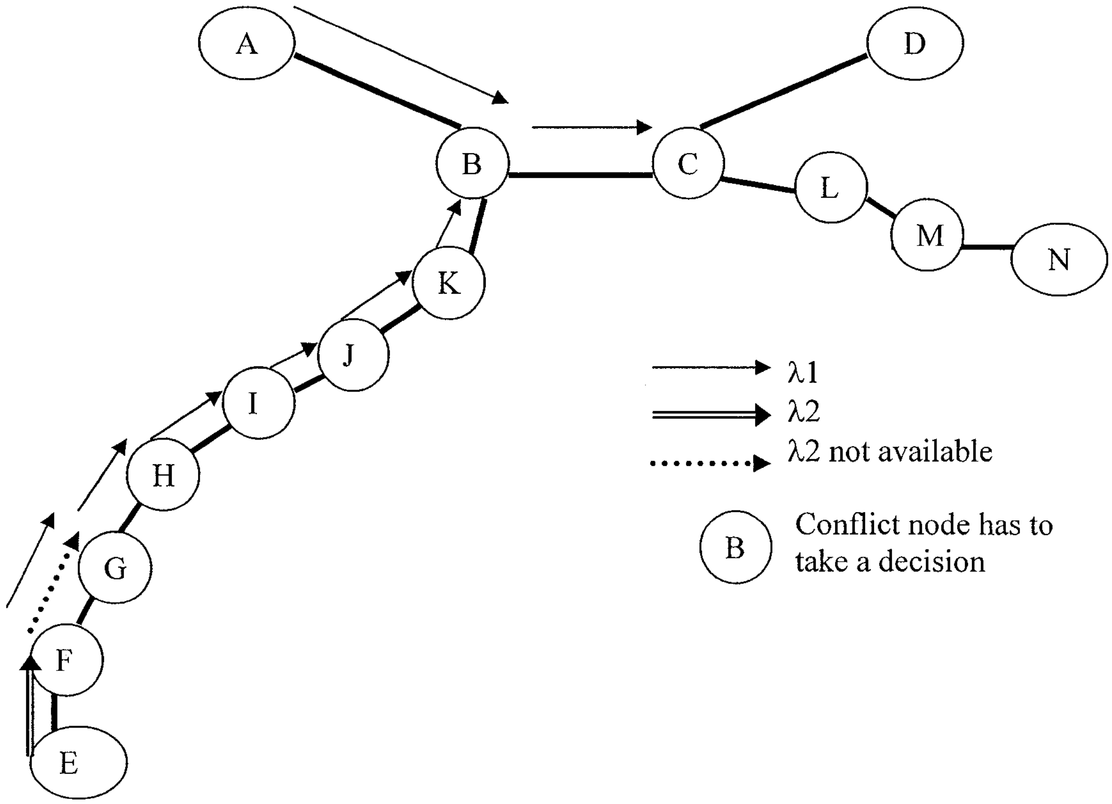


Figure 3.3 Bottleneck Congestion Avoidance optical network example

Req1 := ConReqA, Req2 := ConReqE

N (ConReqA) := 4, N (ConReqE) := 12

F (ConReqA) := 2, F (ConReqE) := 8

S (ConReqA) := 2, S (ConReqE) := 7

In case when a conflict node receives more than two connection requests, it puts the requests in FIFO queue and applies the proposed algorithm on the first two arrival requests, and then it applies the proposed algorithm on the winner request among them with the third arrival connection request and so on till the last connection request in the queue.

3.2 Summary

In this Chapter, we proposed the Bottleneck Congestion Avoidance algorithm to enhance the use of label suggestion provided by RSVP-TE signaling for optical networks. This algorithm makes it possible for more efficient utilization of resources in optical networks. This is done by the decision that is taken by core nodes when they receive more than one request for the same wavelength and output port.

We conclude that our proposed algorithm will improve the blocking and the delay probabilities in optical networks. Furthermore, the proposed algorithm removes the limitation of the suggested label approach that we described in Section 2.1.

Chapter 4

4 ROUTING AND WAVELENGTH ASSIGNMENT IN IP OVER WDM

As we illustrated in Sections 2.2, 2.3 and 2.4, although there have been much work done to allocate and accommodate the requested resources through the optical network, successful routing and labeling assignment could not be always guaranteed unless reasonable amount of signaling takes place. This is especially if the network core nodes are wavelength conversion incapable.

In this Chapter, we first introduce the problem and the challenges in wavelength labeling assignment. Then, we solve the problem by the proposed traffic engineering technique. This technique uses explicit routing to establish the route, the proposed Proper Suggested Label Selection (PSLS) table to distribute the network resources and to help in fast wavelength allocation, and the proposed Lambda Blocking Check Algorithm (LBCA) that inexpensively and efficiently checks resources availability through the optical network. The proposed technique enhances the network utilization and reduces the blocking probabilities to its lowest degrees. Moreover, its significance comes from its signaling cost efficiency that other centralized traffic engineering server techniques lack of.

4.1 Challenges in Wavelength Labeling Assignment and Suggested Label Selection

Selections of suggested labels by ingress nodes could play a nontrivial role in blocking probability. Let us consider the optical network example illustrated in Figure 4.1. Assume the number of available fibers between each node pair in the network is one. Assuming a connection has been requested (Req1) from *A* to *L* through the bottleneck *I* and *K* using suggested label λ1 and another connection has been requested (Req2) from *P* to *O* through *D*, *G*, *H*, *J*, *I*, *K*, *M*, and *N*, using the same suggested label λ1. Conflict in the bottleneck [22] between *I* and *K*, if node *I* receives both the requests (Req1, Req2) before a reservation process occurs for any of Req1 or Req2, can easily be resolved using the solution mentioned in [28] and Chapter 3. However, such a solution could be on the expense of one of the connections, for example, in case of readjustments needed to previous adjusted nodes that may cause nontrivial delay in the connection. The solution could also be on the expense of the entire connection request, for example, when a connection must be blocked to allow other connection to proceed (if resources, wavelength, are no more available).

Such a problem will be much more obvious if the rest of the ingress nodes wish to establish their connections through the same bottleneck *I* and *K* using the same wavelength λ1. For example, if Req3 is the connection request from ingress *B* to egress *L*, through *E*, *I*, and *K* nodes using λ1, and Req4 is the connection request from ingress *C* to egress *L*, through *F*, *I*, and *K* using λ1. Assume Req1 is received by node *I* first whereas Req2, Req3 and Req4 are received by node *I* at the same time and after Req1 but no reservation acknowledgment has been made yet for Req1. In such a case, λ1 will not be available in the bottleneck *I*, *K* for more than one request. Req2 will win the reservation based on the "Bottleneck Congestion Avoidance" algorithm described in Section 3.1, where administrator concern is to reduce the delay probability rather than blocking probability and assuming that the suggested label passes successfully through the path from *P* to *I* nodes. All other requests (Req1, Req3 and Req4) must be failed and reestablished (reinitiated). This adds a nontrivial delay to their connectivity. Moreover, the conflict might happen again through the bottleneck or farther, whenever the ingress nodes attempt to use the same wavelength again in their requests.

In order to solve the above discussed problem properly and in signaling cost efficient manner, we propose a new traffic engineering technique in Section 4.2.

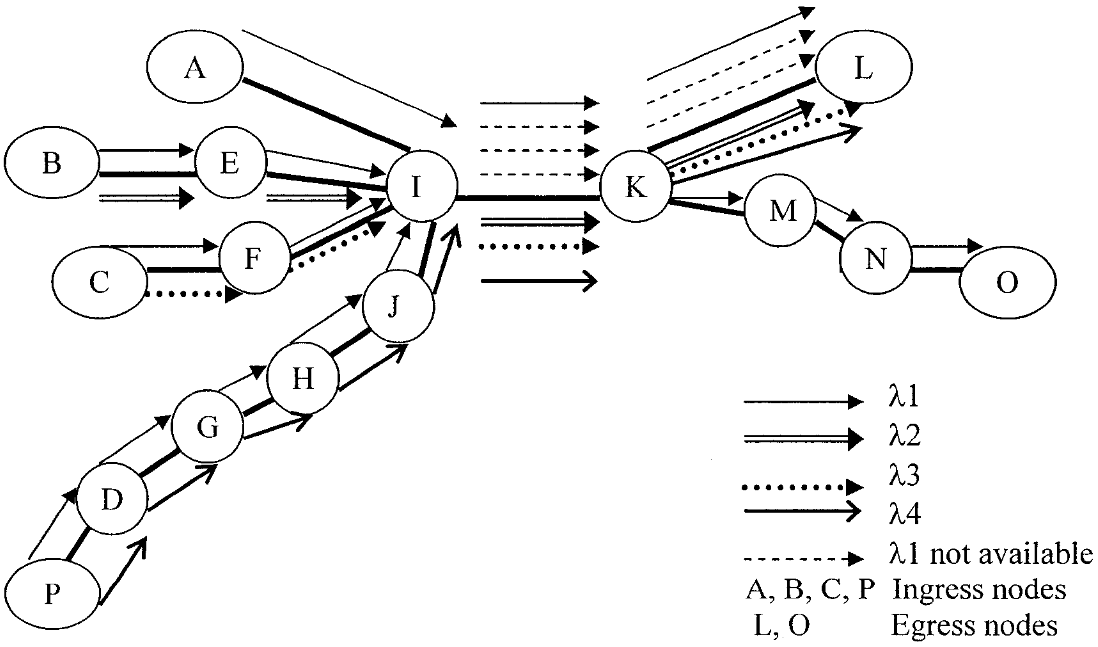


Figure 4.1 Optical network example

4.2 The Proposed Traffic Engineering Technique

The proposed technique gives a Traffic Engineering (T-E) server the right to control and distribute the routes and the wavelengths (λs) in the optical network. It assumes that all ingress nodes are numbered from *0* to *I-1* where *I* is the number of ingress nodes in the network. Moreover, it assumes that the number of fibers between each pair in the network is known to the T-E server, and the core nodes are wavelength conversion incapable.

4.2.1 Routing and Wavelength Assignment

When ingress node *i* requests a connection (λ or range of λs depending on its need and authorization) with egress node *e* from T-E server, the T-E server uses explicit routing to search for a route from already established routing table defined through a routing mechanism (for example, OSPF [29]), and selects the appropriate route (if available) from ingress *i* to egress *e*. If there is no available route, it will fail the request and update the ingress *i* with the result of its request.

After a successful route selection, T-E server calls the Proper Suggested Label Selection (PSLS) algorithm, described in Section 4.2.2, to suggest λ (or λs) for the request depending on the PSLS table. It then calls the Lambda Blocking Check Algorithm (LBCA), described in Section 4.2.3, to check if the suggested λ (or λs) are available through the entire selected route. If the suggested λ (or λs) is (are) not available through the selected route, T-E server will keep trying to find new λs by the PSLS and test their availability by LBCA. If there is no more λs to be suggested by the PSLS, T-E server will try to accommodate a new route for the request and to accommodate the suggested λs through the route. If no more new routes are available, it will fail the request.

4.2.2 The Proper Suggested Label Selection Algorithm

The PSLS algorithm is used by the T-E server to distribute the labels between ingress nodes, and so within the optical network. It also minimizes the congestion cost through the network and its bottlenecks [28]. By this, it speeds up the T-E server search for the requested resources. PSLS generates a table (as shown in Table 4.1) for the network that contains a column for each ingress node. Each column has IDs that indicate the priority of the ranges from where the suggested label must be selected for each individual ingress node. The ranges can easily be obtained by filling the IDs in the Range of Each Ingress Lambda Selections array (REILS (ID)) as described below. When an ingress node requests a connection, the label (or range of labels) that must be tried to be suggested and accommodated is (are) from the top of the ingress's column down to its bottom. PSLS table must be generated only once. However, it must be regenerated again every time the network topology changes.

Given that
- I: number of Ingress nodes in the network
- LN: Number of Lambdas available in the network
- RA: Range Average of wavelength allocated for ingress nodes in the network;
  RA = trunc (LN / I)
- RRA: Rest of RA;
  RRA = (LN%I)
- REILS(i): array of Range of Each Ingress *i* Lambda Selections;
  REILS(i) = [(RA * i), (RA * (i + 1)) - 1]
  where i = {0, 1, 2,...., I-1}
- REILS (I): Last REILS, if RRA > 0;
  REILS (I) = [(RA * I), (RA * I) + (RRA - 1)]
- $ID_{j,i}$: ID table selection for ingress nodes *i* and priority *j* to find REILS ($ID_{j,i}$) where

$ID_{j,i} = \{0, 1, 2, \ldots\ldots\ldots, I-1\}; i, j = \{0, 1, 2, \ldots\ldots, I-1\}$

- CN: Central ingress Node;
  CN = trunc (I/2)

The PSLS table can be generated in five steps as shown in the pseudo code in Figure 4.2.

```
// Five steps coding to generate the PSLS table (ID [Priority (j)] [Ingress (i)]).
// First step
Ingress (i) = CN;
{
   find the priority IDs of Ingress (i) from the Priority (j) = 0 till the Priority (j) = I-1
   where the priority ID corresponding to REILS (ID) begins from the ingress CN itself,
   down to the rest ingress nodes starting from the farthest ingress node toward the
   closest ingress node;
}

// Second step
for (Ingress (i) = 0; Ingress (i) < I; Ingress (i) ++)
{
   if (Ingress (i) != CN)
   {
      find the priority IDs of Ingress (i) for the Priority (j) = 0, I-1, I-2;
   }
}

// Third step
for (Ingress (i) = CN+1; Ingress (i) < I; Ingress (i) ++)
{
   find the priority IDs of Ingress (i) from the Priority (j) = 1 till Priority (j) = I-3;
}

// Fourth step
Ingress (i) = 0;
{
   find the priority IDs of Ingress (i) from the Priority (j) = 1 till Priority (j) = I-3;
}

// Fifth step
for (Ingress (i) = 1; Ingress (i) < CN; Ingress (i) ++)
{
   find the priority IDs of Ingress (i) from the Priority (j) = 1 till Priority (j) = I-3;
}
```

Figure 4.2 The Proper Suggested Label Selection Table pseudo code

First step: find for i = CN (Central Node), all j = 0, 1,..., I-1

Ingress(i) Priority (j)	Ingress (0)	Ingress (1)	Ingress (2)	Ingress (3)
First			2	
Second			0	
Third			3	
Fourth			1	

Second step: find for all i (except i = CN), j = 0, I-1, I-2

Ingress(i) Priority (j)	Ingress (0)	Ingress (1)	Ingress (2)	Ingress (3)
First	0	1	2	3
Second			0	
Third	1	2	3	0
Fourth	3	0	1	2

Third step: find for i >= CN+1, j = 1,...., I-3

Ingress(i) Priority (j)	Ingress (0)	Ingress (1)	Ingress (2)	Ingress (3)
First	0	1	2	3
Second			0	1
Third	1	2	3	0
Fourth	3	0	1	2

Fourth step: find for i = 0, j = 1,...., I-3

Ingress(i) Priority (j)	Ingress (0)	Ingress (1)	Ingress (2)	Ingress (3)
First	0	1	2	3
Second	2		0	1
Third	1	2	3	0
Fourth	3	0	1	2

Fifth step: find for CN - 1>= i > =1, j = 1,...., I-3

Ingress(i) Priority (j)	Ingress (0)	Ingress (1)	Ingress (2)	Ingress (3)
First	0	1	2	3
Second	2	3	0	1
Third	1	2	3	0
Fourth	3	0	1	2

Table 4.1 The Proper Suggested Label Selection Table example

Let us consider the same network example illustrated in Figure 4.1, and number the ingress nodes as: *A* = 0, *B* = 1, *C* = 2, and *P* = 3 where *I* = 4; *LN* = 4; *RA* = 1; *RRA* = 0; *CN* = 2; *REILS* (0) = [0, 0]; *REILS* (1) = [1, 1]; *REILS* (2) = [2, 2]; *REILS* (3) = [3, 3]; *REILS* (4) does not exist. Note that we assume that the wavelengths always count from *0* to *LN-1* and so λ1, λ2, λ3, and λ4 in Figure 4.1 are equivalent to λ0, λ1, λ2, and λ3 respectively. Using the Proper Suggested Label Selection and its pseudo code illustrated in Figure 4.2, we can find the PSLS table as illustrated in Table 4.1. Using the PSLS table, the suggestion label for an ingress request will be first suggested from the REILS of the ingress itself, then from the REILS of the rest ingress nodes starting from the farthest ingress node toward the closest ingress node.

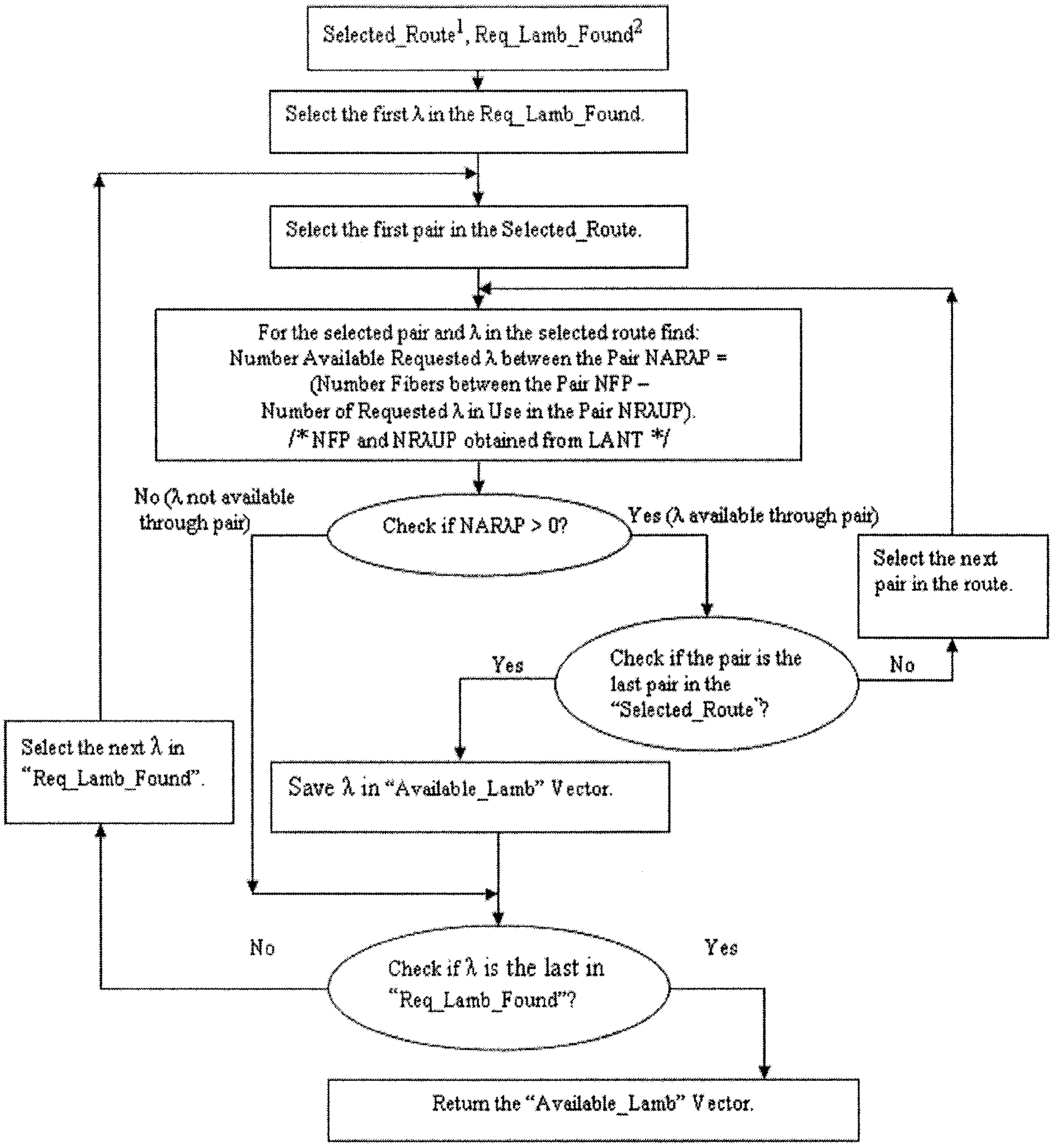
4.2.3 Lambda Blocking Check Algorithm

T-E server uses LBCA to check the availability of the label (or range of labels) through the route by obtaining information from Lambda Availability through the Network Table (LANT). LANT consists of number of columns equal to the number of the node pairs of the network, and number of rows equal to the Number of Lambdas available in the network plus one. For example, the network illustrated in Figure 4.1 has the LANT shown in Table 4.2 where NFP values indicate the Number of Fibers installed between each of the Pairs. λs values indicate the usage capacity of each λ in the pairs. LANT must be updated whenever any connection is added to or removed from the network by the T-E server. For example, if a connection between ingress *A* and egress *L* has been successfully established through nodes *I* and *K* using λ0, each pair of the route (*A-I*, *I-K*, *K-L*) must be updated by incrementing the corresponding selected label value of the pair by one (λ0 in this case). On the other hand, when a connection tears down, updating is by decrementing the corresponding selected label value of the pair by one.

The working mechanism of LBCA is illustrated in Figure 4.3. LBCA takes the selected route and the label (or range of labels) array needed to be checked as its input. It selects the first λ in the requested labels array and the first node's pair in the selected route. Then, for the selected pair and λ in the selected route, it finds Number Available Requested λ between the Pair (NARλP) from LANT. Assuming NFP equal to the Number of Fibers between the Pair and NRλUP equal to the Number of Requested λ currently in Use in the Pair, LBCA easily finds NARλP = NFP – NRλUP. NARλP>0 indicates that selected λ is available in the selected pair and so, LBCA continues the blocking free check for the λ till the end of the selected route. If λ is available through the entire route, it will be saved in a vector and the blocking free check will continue for the rest of λs in case more than one λ is needed to be checked. NARλP = 0 indicates that selected λ is not available in the selected pair and so, it is not available through the selected route of the connection request. When the blocking free check completes for the label (or range of labels), LBCA returns the check result.

Node Pairs λs + NFP	A-I	I-K	K-L	K-M	M-N	N-O	B-E	E-I	C-F	F-I	P-D	D-G	G-H	H-J	J-I
λ0															
λ1															
λ2															
λ3															
NFP															

Table 4.2 **Lambda Availability through the Network Table example**



Terms used in the Figure:

1. "selected_route": successfully selected route by the T-E server for a connection request.
2. "Req_Lamb_Found": Array of suggested labels requested to be found and checked for blocking free through the selected route.

Figure 4.3 Lambda Blocking Check Algorithm

4.3 Summary

In this Chapter, we introduced some challenges and problems in labeling assignment. In order to solve these problems, we proposed the Traffic Engineering Technique that uses the explicit routing to determine the routes between ingress and egress nodes. This technique uses the proposed Proper Suggested Label Selection Algorithm to distribute the network resources, and the Lambda Blocking Check Algorithm that easily and efficiently checks the resources availability through the optical network. The proposed technique makes it possible and real to have a blocking free labeling decision for the connection requests in the optical network, where the network core nodes are wavelength conversion incapable.

In the next Chapter, we will demonstrate the proposed T-E technique efficiency by showing its cost requirement and by comparing the blocking probability of the proposed technique with the OSPF through simulations.

Chapter 5

5 PERFORMANCE EVALUATION AND SIMULATION

. In this Chapter, first, we demonstrate the efficiency of the proposed T-E technique (as described in Chapter 4) by showing its cost requirement. Then we build an optical network model with respect to software engineering and object oriented point of view to illustrate the efficiency of the proposed traffic engineering technique by simulation results.

5.1 Cost Requirement

The cornerstone of the proposed T-E technique cost efficiency is LBCA mechanism. By using the LBCA mechanism, core nodes do not have to update their states to the T-E server. Only the ingress nodes need to update their states to the T-E server. This is what makes the proposed technique unique among other centralized T-E techniques.

Here we evaluate our proposal with respect to the number of messages required and memory required in the T-E server.

The Total Number of Messages (TNM) required to suggest *N* successful connection requests and to update the states is given by:

TNM = 3 * N

where each successful connection request requires the three following messages:

- IngConReq: Ingress Connection Request message.
- T-E_Resp: T-E Response message.

- IngConAck: Ingress Connection Acknowledgment message.

On the other hand, the core nodes need to locally assign the wavelengths between their inputs and outputs.

The Memory Required (MR) in the T-E server is given by the following. There is no memory required in the ingress, core, or egress nodes.

MR = ER + PSLST entries + LANT entries

where

- ER: number of the Explicit Routes in the network;
- PSLST entries = pow (2,I)

  where

  I: number of Ingress nodes in the network;
- LANT entries = (LN + 1) * NumPairs

  where

  LN: Number of Lambdas available in the network;

  NumPairs: Number of node Pairs in the network.

5.2 Simulation

To illustrate the performance efficiency of the proposed traffic engineering technique, we first need to build a network model. For this purpose, we use the UML language and the Rational Rose modeling tool. Second, we use Java as the programming language to perform the simulation and analyze the result.

5.2.1 Building the Model

In this section, we build an optical network model with respect to software engineering and object oriented point of view to be able to simulate the proposed traffic engineering technique performance. As the Class Diagram model illustrates in Figure 5.1, fifteen Classes (including the Application Class) are used to build the model. We can divide the model into the following three sub-models:

First, the Network Nodes those are in charge of handling the connection requests. Second, the Connection Request Object which has the connection requirements and will be sent between the network nodes while recording their responses to the request. Third, the Proposed T-E Technique Model that will be in charge of simulating our proposed T-E Technique. Those three models are described in more details in the following subsections.

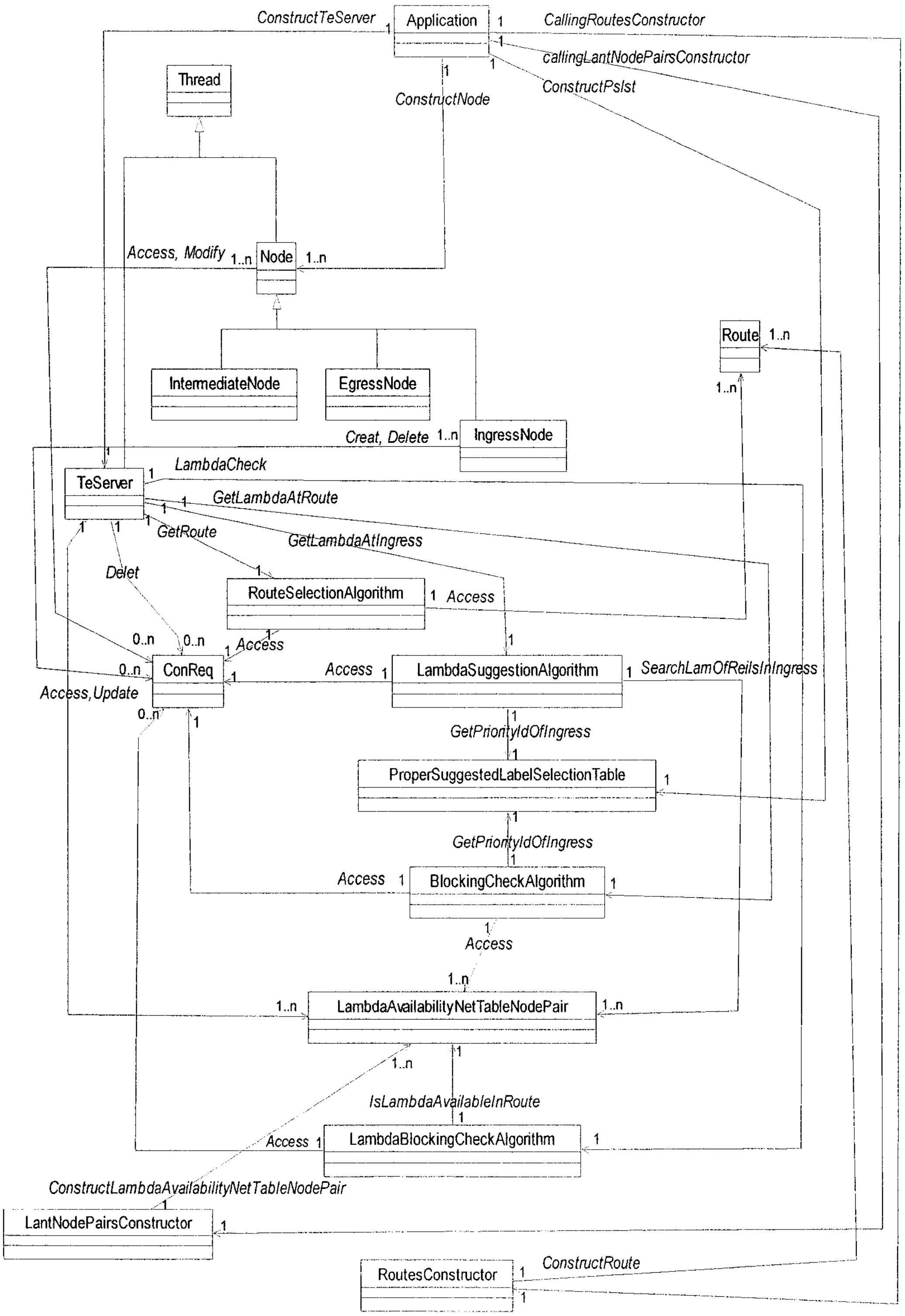


Figure 5.1 Class Diagram model

5.2.1.1 Network Nodes

We split the network nodes into three types. First type is the ingress nodes (IngressNode Class). Second type is core nodes (IntermediateNode Class). Third type is the egress nodes (EgressNode Class). To perform a real time simulation, the three types (Classes) are inherited from Thread Class.

The IngressNode Class is mainly responsible for creating and requesting connections from the T-E server (described in Section 5.2.1.3). Depending on the T-E response, if the connection request is approved for further processing, ingress node proceeds in processing the connection request by handling the connection request object to the next node in the selected route. If the ingress node receives reservation acknowledgment for the connection request from rest of the nodes in the selected route, it acknowledges the end of reservation success by reserving the resources and reporting the connection request reservation success to the T-E server. At this point, the connection is successfully established. Otherwise, in case that the reservation fails in any of the nodes or the connection request times out before receiving the reservation acknowledgment, the ingress node reports the failing results to the T-E server. If from the beginning the connection request is not approved by the T-E server for further processing because of unavailable resources, the unapproved connection request will be considered as a failure to the request by the ingress node.

The IntermediateNode Class is responsible for updating the connection request object internally depending on the intermediate node resources availability.

The EgressNode Class has the same responsibilities of the IntermediateNode Class. In addition, the egress node plays the role of egress and ingress node at the same time. The

UML models of the IngressNode Class, the IntermediateNode Class and the EgressNode Class are illustrated in Figures 5.2, 5.3 and 5.4 respectively.

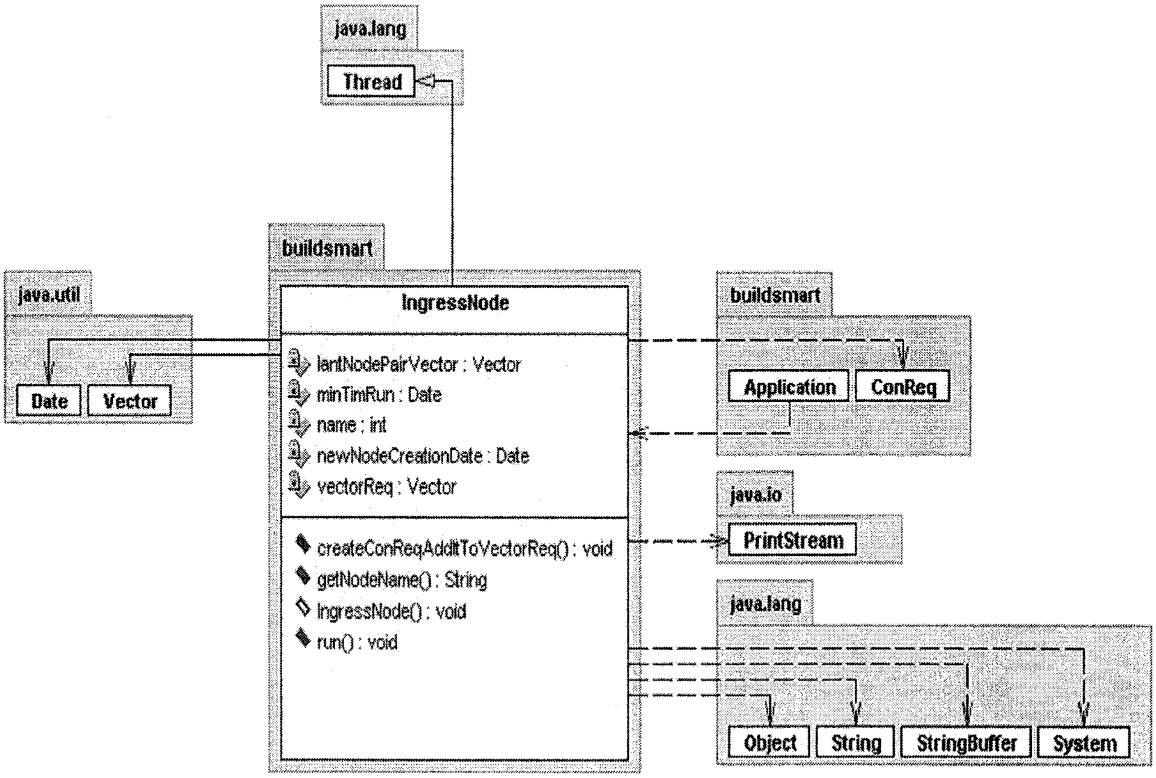


Figure 5.2 IngressNode Class

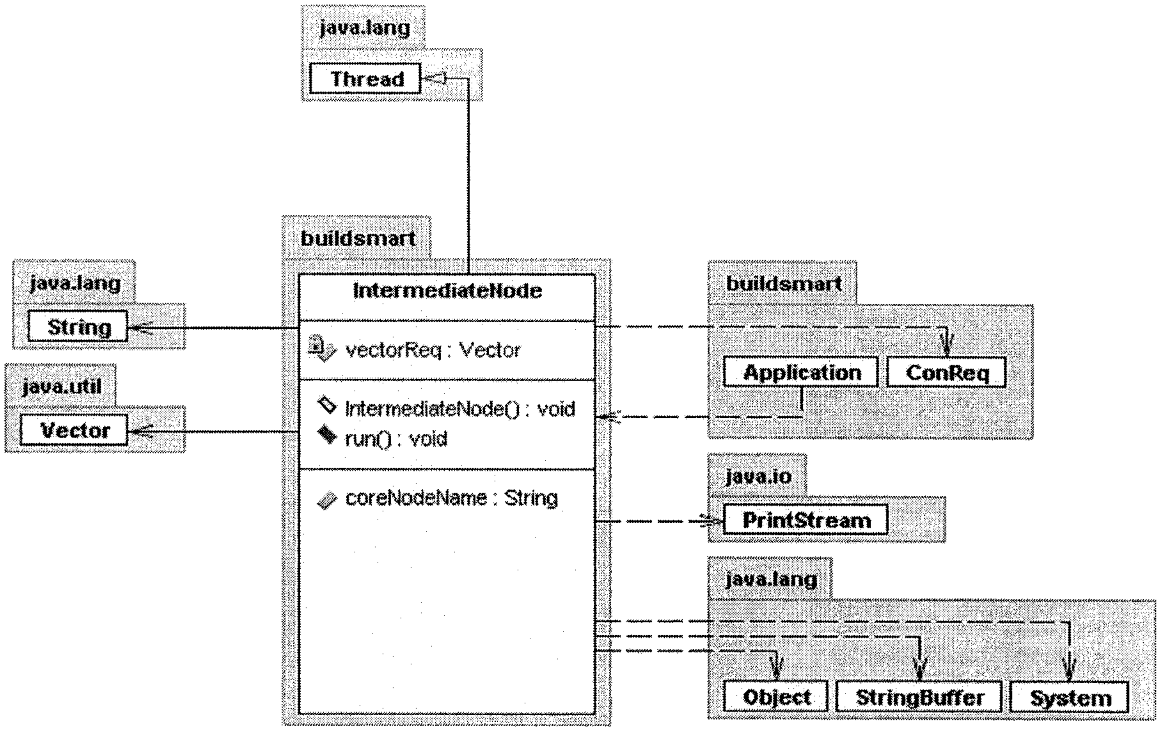


Figure 5.3 IntermediateNode Class

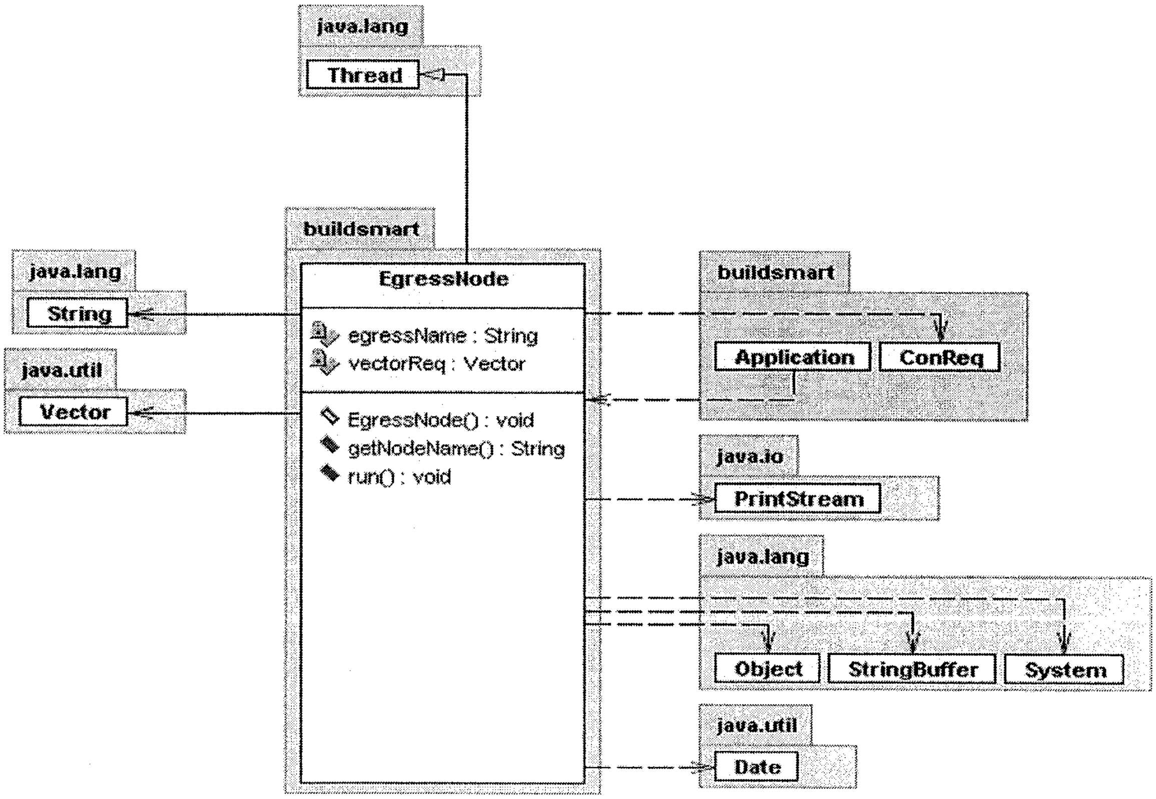


Figure 5.4 EgressNode Class

5.2.1.2 Connection Request Object

The connection request object (implemented by ConReq Class) is the cornerstone of demanding, processing, ending and tearing a connection. By updating the status of a connection request (implemented by statusConReq array), the network nodes and the T-E server can determine the state of the connection. Depending on its state they process the connection request further. The ConReq object fields are: “serverCheck”, “tear”, “procConDiscon” and “procDiscon”.

The “statusConReq” array pseudo code and fields:

```
private int [] statusConReq = new int [4];
statusConReq: int [serverCheck, tear, procConDiscon, procDiscon]
```

Each field’s available values and states interpretation:

"severCheck" = 0 --- New Connection Request.

"severCheck" = 1 --- Success to find the route, Lambdas at Ingress and Lambdas in the entire route.

"severCheck" = 2 --- Fail to find the route and/or Lambdas at Ingress and/or Lambdas in the entire route.

"tear" = 0 --- No tear request.

"tear" = 1 --- tear request.

"procConDiscon" = 0 --- The "ConReq" must be served by the server or/and the nodes for a new connection request (usually by ingress), to be checked by T-E server, then by the nodes.

"procConDiscon" = 1 --- The "ConReq" has been served by the T-E server and the nodes regarding establishing a connection. No need for further processing unless it sets to 0.

"procDiscon" = 0 --- The "ConReq" must be served by the server or/and the nodes for a tear request to be processed by the nodes, then by the T-E server.

"procDiscon" = 1 --- The "ConReq" has been served by the T-E server and the nodes regarding tearing a connection. No need for further processing unless it sets to 0.

The UML model of the ConReq Class is illustrated in Figure 5.5.

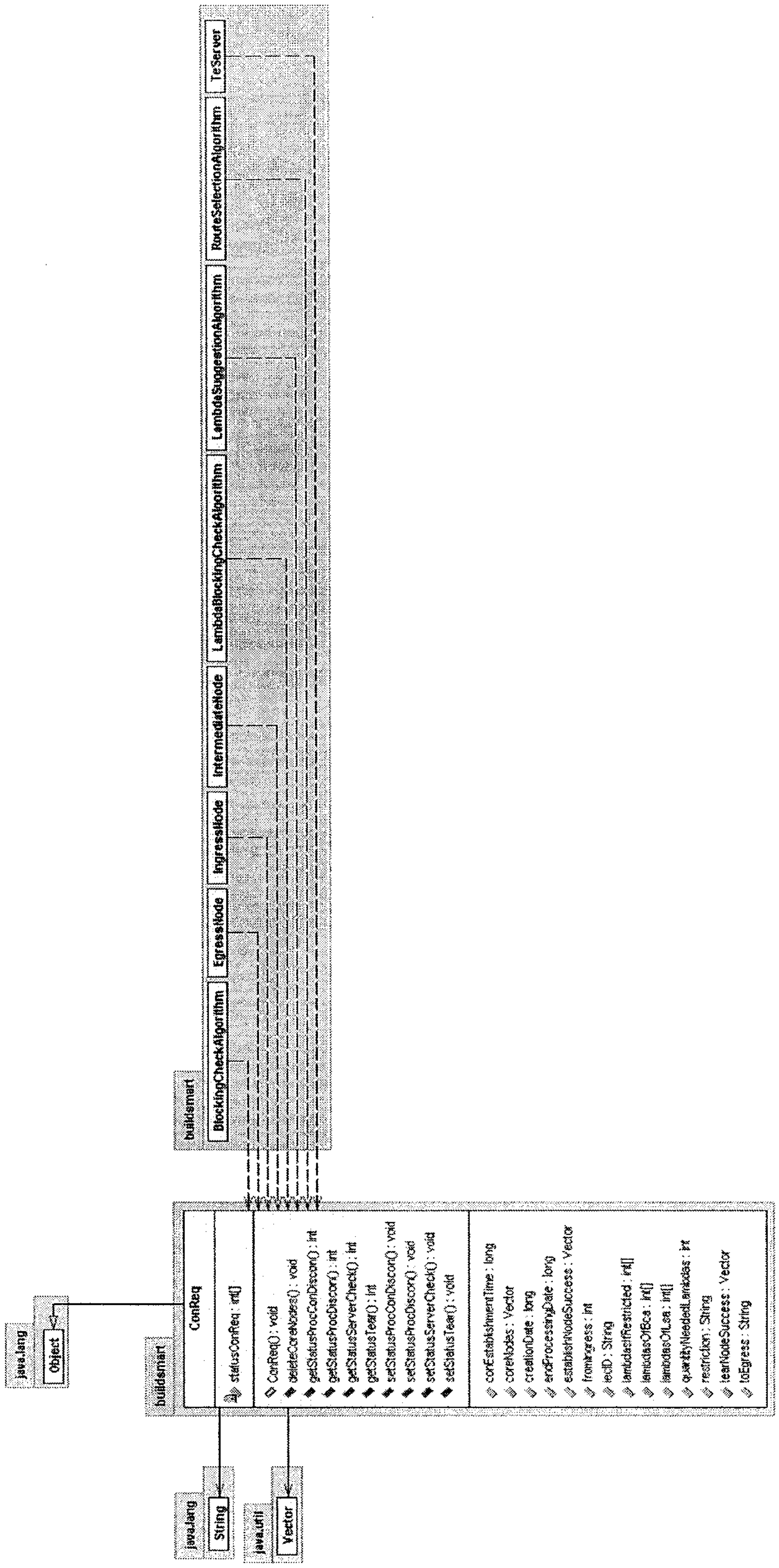


Figure 5.5 ConReq Class

5.2.1.3 The Proposed T-E Technique Model

The proposed T-E technique model is mainly built from the T-E server model, Route Selection Algorithm model, Lambda Suggestion Algorithm model, Proper Suggested Label Selection Table model, Lambda Blocking Check Algorithm model, Blocking Check Algorithm model and Lambda Availability Net Table Node Pair model.

The T-E server (TeServer Class) is the cornerstone of the proposed T-E technique. It controls rest of the models and coordinates interactions between them. As the sequence diagram for a successful connection request in Figure 5.6 illustrates, when ever a connection is requested by an ingress node from the TeServer, the TeServer requests a route from the RouteSelectionAlgorithm Class by sending the RouteRequest message. The RouteSelectionAlgorithm is responsible for returning the best shortest route for the requested connection using RouteReturn message. When the route is available, the T-E server uses the LambdaSuggestionAlgorithm Class to get the suggested resources for the connection by sending the RequestForLambda message. The LambdaSuggestionAlgorithm Class requests a suggested range from the ProperSuggestedLabelSelectionTable Class, then it returns the suggested lambda to the TeServer. The TeServer checks the resource availability by using the LambdaBlockingCheckAlgorithm and the BlockingCheckAlgorithm classes that are responsible for checking the resource availability through the entire selected route by accessing the LambdaAvailabilityNetTableNodePair Class. When the TeServer receives ReturnCheckResult message, it updates the LambdaAvailabilityNetTableNodePair and returns the selected route with the suggested resources to the IngressNode Class. The IngressNode sends a path message toward the EgressNode and through the

IntermediateNodes. The EgressNode sends the ReservationMessage toward the IngressNode. The IngressNode sends ConReqReservationAcknowledgment message to the server. It is important to mention that the TeServer Class is inherited from Thread Class to perform a real time simulation.

The TeServer, RouteSelectionAlgorithm, LambdaSuggestionAlgorithm, ProperSuggestedLabelSelectionTable, LambdaBlockingCheckAlgorithm, BlockingCheckAlgorithm and LambdaAvailabilityNetTableNodePair Classes are illustrated in Figures 5.7, 5.8, 5.9, 5.10, 5.11, 5.12, and 5.13 respectively.

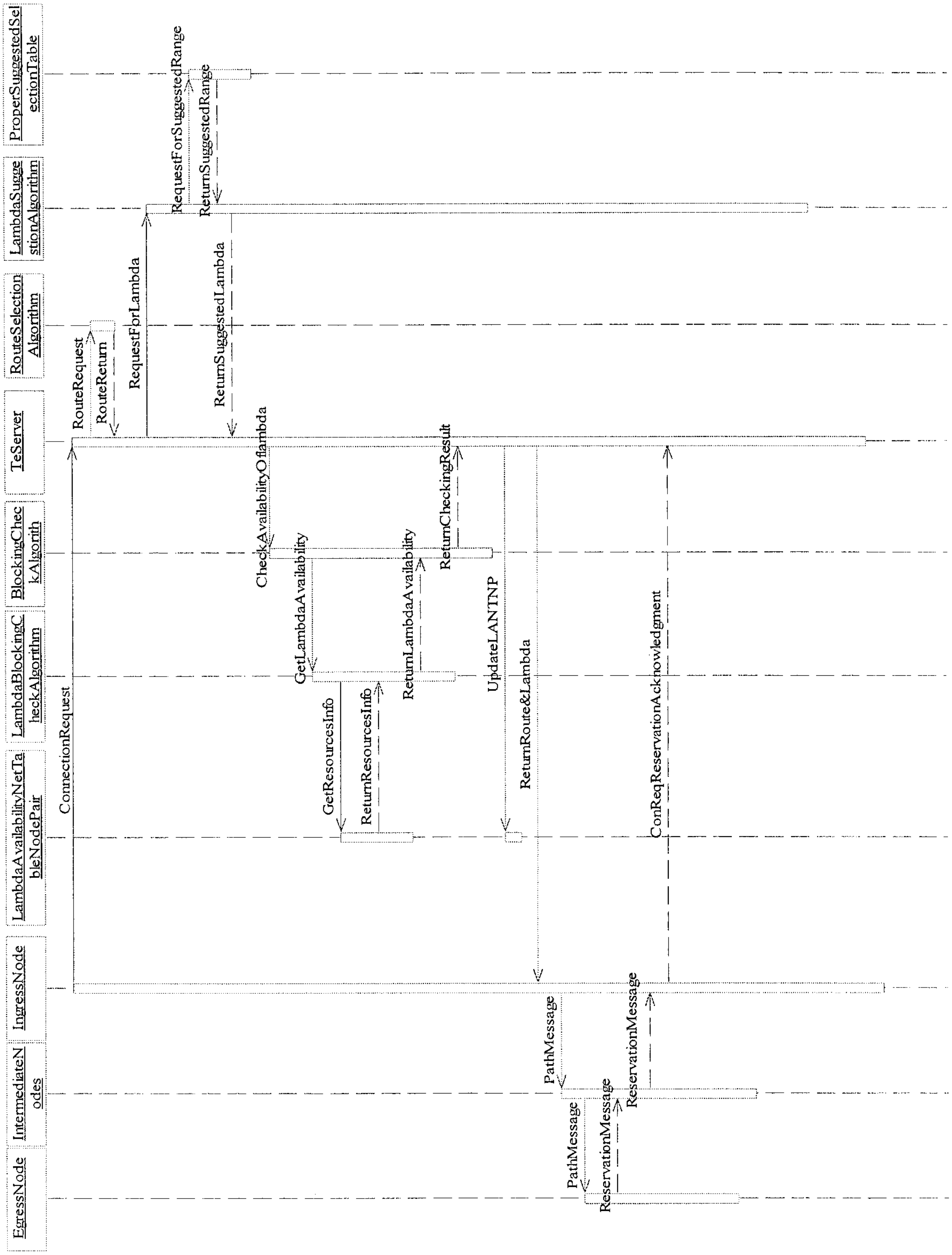


Figure 5.6 Sequence Diagram for a successful connection request

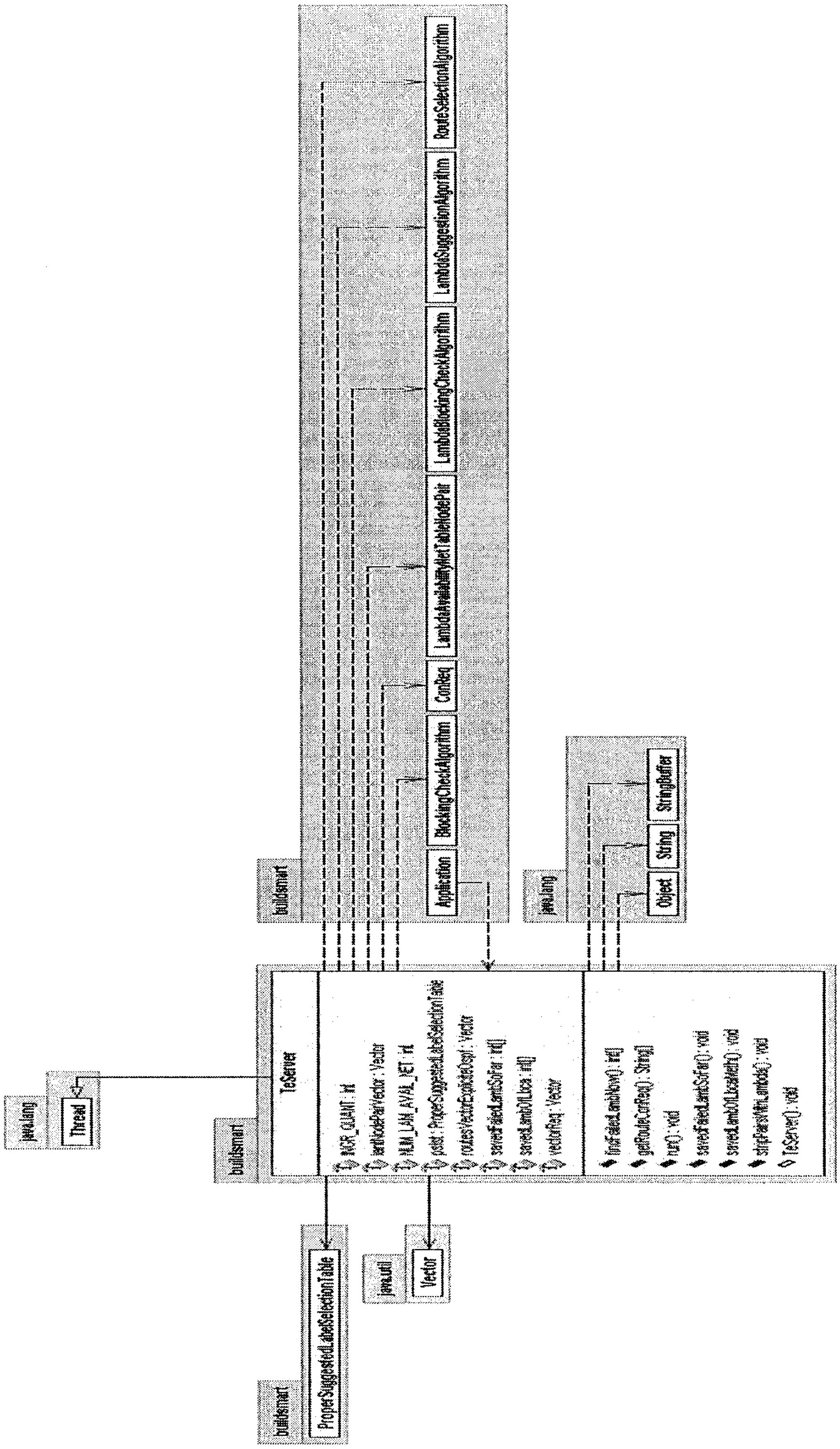


Figure 5.7 TeServer Class

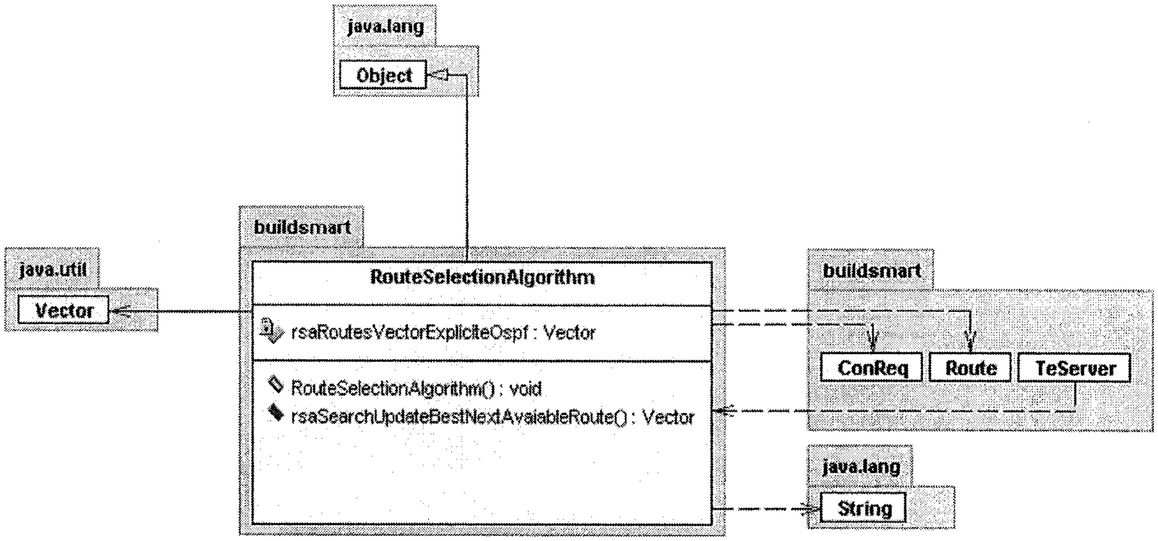


Figure 5.8 RouteSelectionAlgorithm Class

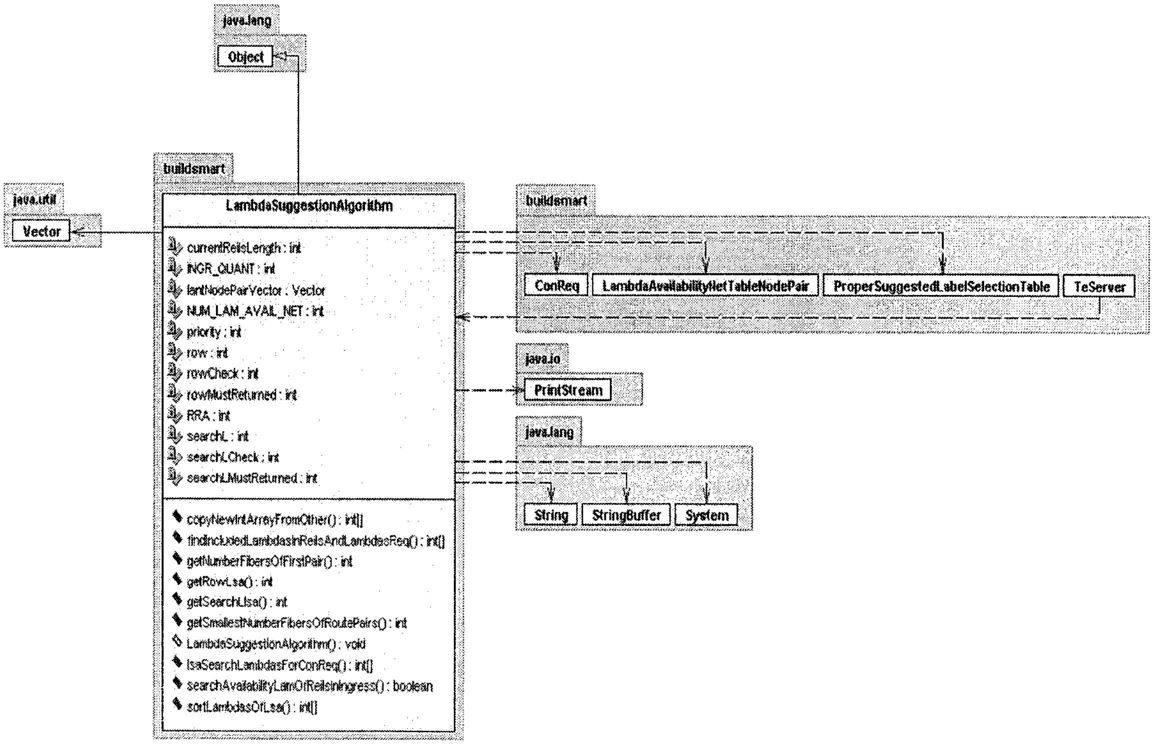


Figure 5.9 LambdaSuggestionAlgorithm Class

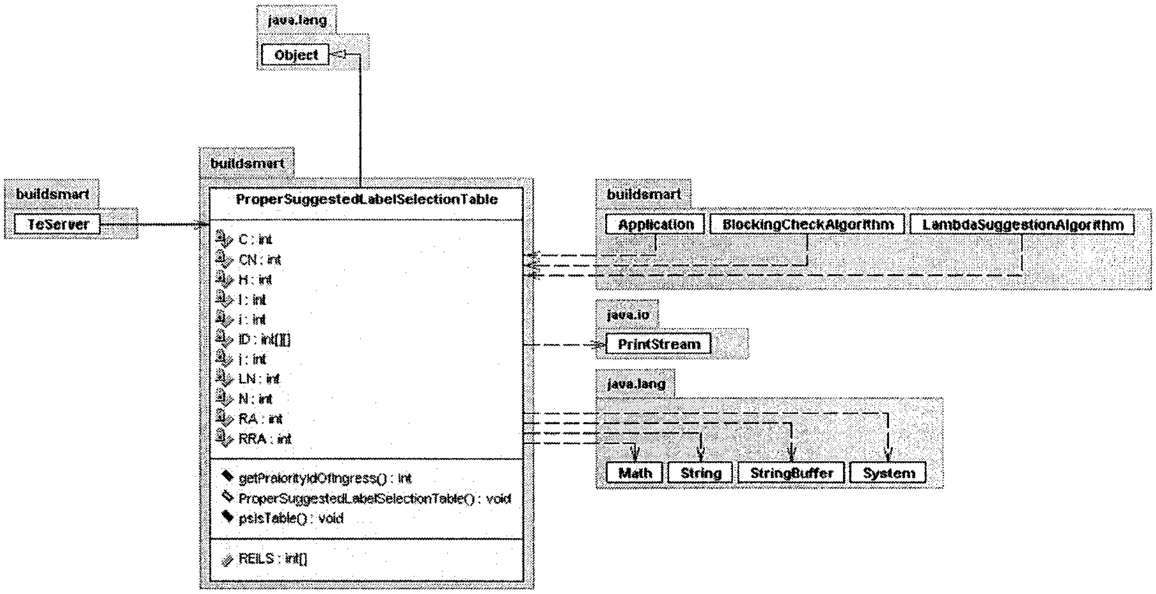


Figure 5.10 ProperSuggestedLabelSelectionTable Class

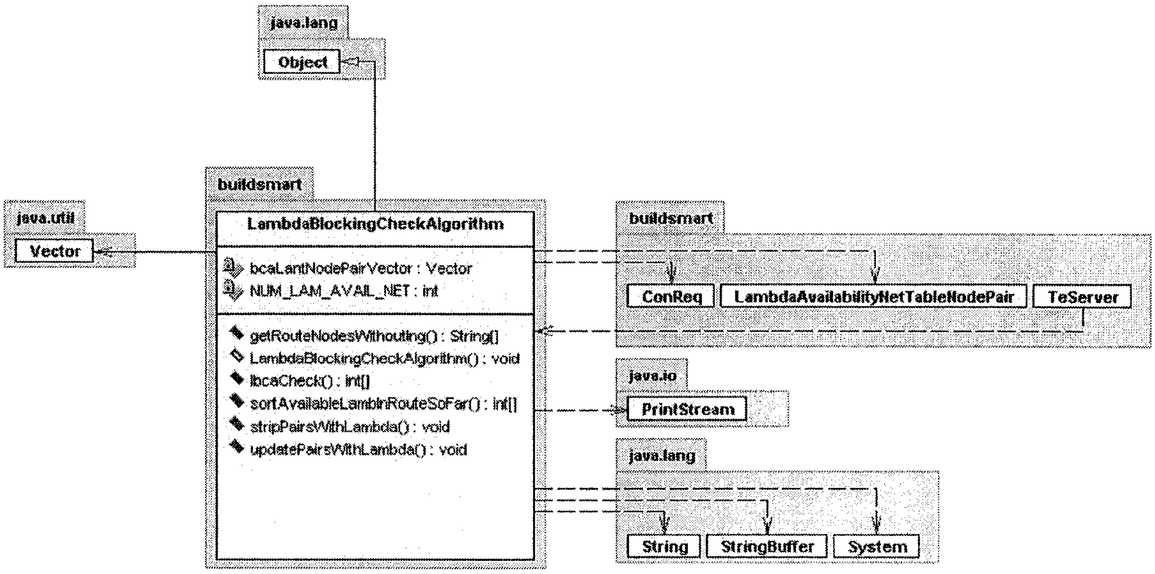


Figure 5.11 LambdaBlockingCheckAlgorithm Class

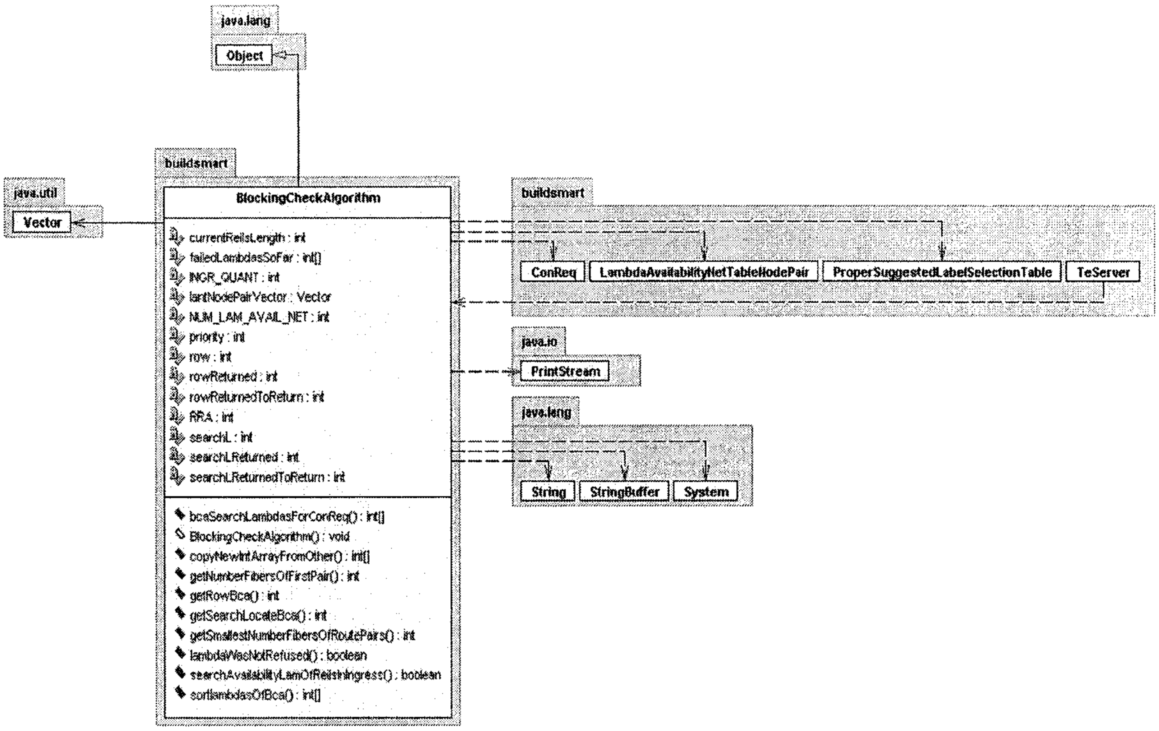


Figure 5.12 BlockingCheckAlgorithm Class

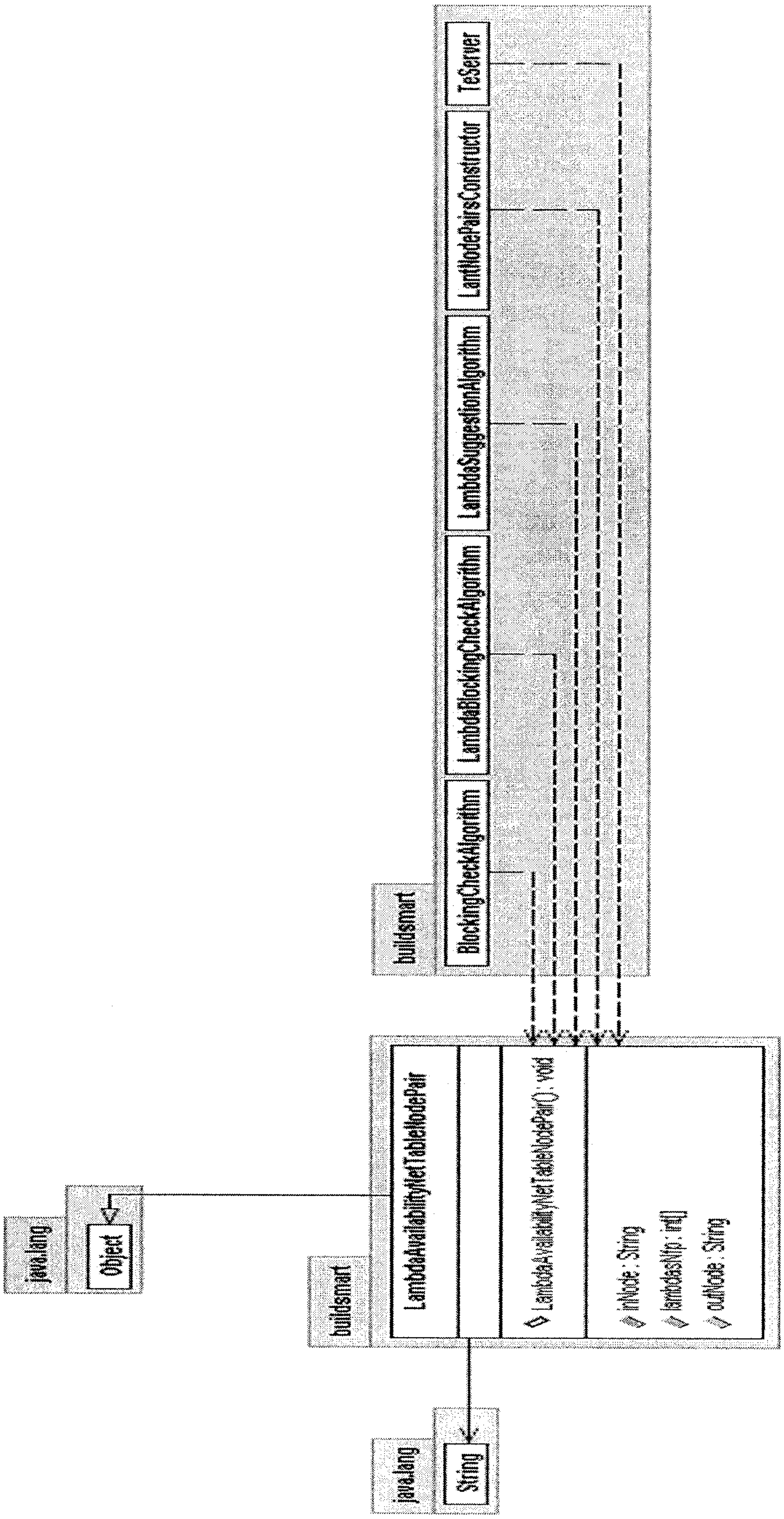


Figure 5.13 LambdaAvailabilityNetTableNodePair Class

5.2.2 Simulation and Results

The performance of the proposed Traffic Engineering (T-E) technique and that of the OSPF algorithm have been analyzed using simulation. The simulation tests were carried out on the optical network model shown in Figure 4.1. The maximum wavelength available for use on each link is four (LN = 4). For this topology model, two cases were tested. For both tests, the connection requests for λs are distributed fairly between ingress nodes. However, the destination egress node for each request is generated randomly. The number of simulation runs for both tests is hundred times continuously. The Number of Fibers installed between each of the network node Pairs (NFP) is one for Test 1. NFP is also one for Test 2 except in the bottleneck *I-K* where it is two. For the first case (Test 1), each simulation run involves four Connection Requests (ConReq). For the second case (Test 2), each simulation run involves eight ConReq. The parameter values for both tests are summarized in Table 5.1 for a single simulation run. Therefore the Total Connection Requests (TotConReq) for Test 1 are four hundred requests and for Test 2 are eight hundred requests.

Figure 5.14 and Figure 5.15 show the blocking performance simulation results for Test 1 and Test 2 respectively. We observe that improvement in the blocking probability is achieved by using our proposed T-E technique. As shown in Figure 5.14, the blocking probability is zero for the proposed technique during all the simulation runs. This means that we were able to utilize the maximum network resources through the bottleneck *I-K*. From the results shown in Figure 5.15, we observe that we were able to maintain a blocking probability improvement by using our T-E technique. Using our proposed technique, blocking occurs in Test 2 whenever more than four connection requests (wavelengths) are requested to the same destination (egress *L* or *O*) at a time. This is due to the fact that between the nodes *K-L* and *K-O*, NFP is one. We can not therefore utilize more than the available four wavelengths. However, we were still able to utilize the entire available wavelengths among the possible routes from a source to a destination. We can conclude that with the use of our proposed algorithms the probability of utilizing the wavelengths in a bottleneck is clearly increasing and remaining within the range [MAR /LN*bottleneck NFP, 1] where bottleneck NFP is the Number of Fibers in the bottleneck node Pair. MAR is the Maximum Available Resources from ingress to egress of the worst route that the bottleneck can be part of; where the worst route is the route that has the node pair with the minimum number of fibers.

Parameter Symbol	Test 1	Test 2
LN	4	4
NFP	1	1
NFP of *I-K*	1	2
ConReq	4	8

Table 5.1 Parameter Values for Test 1 and Test 2

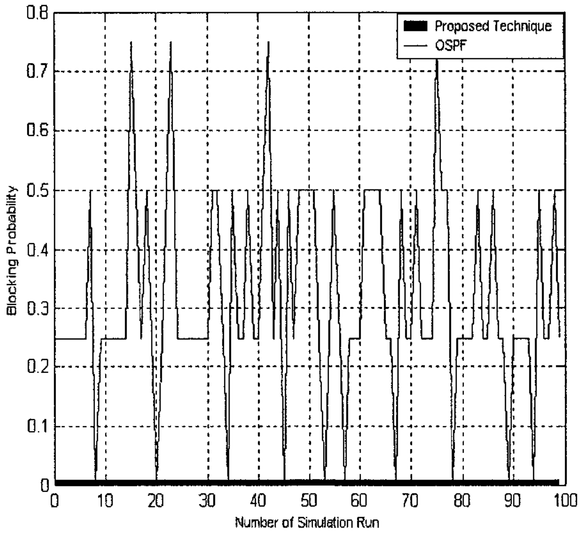


Figure 5.14 Blocking probabilities (Test 1)

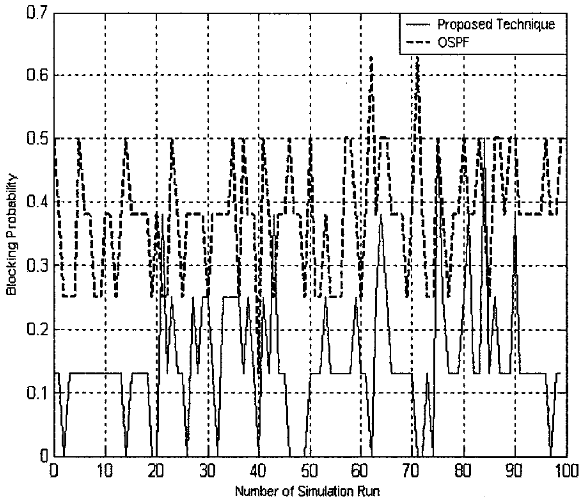


Figure 5.15 Blocking probabilities (Test 2)

5.3 Summary

This Chapter demonstrated the efficiency of the proposed T-E technique. First, we showed its cost efficient way of updating the network nodes states and its low memory requirements. Second, by building an optical network model with respect to software engineering and object oriented point of view, we showed the efficiency of the proposed traffic engineering technique by the simulation results.

Chapter 6

6 CONCLUSION AND FUTURE WORKS

The required high speed data traffic and bandwidth in any domain can be achieved by using the optical network by means of its intelligent architecture. By the development of GMPLS, we are able to eliminate the use of ATM, SONET and their redundancy by moving their functions to the routers, OXCs and DWDM. However, routing and wavelength labeling assignment difficulties occur due to the optical nature where label means much more than just identification for a special traffic. Such difficulties increase especially if the network core nodes are conversion incapable. In such case, if a core node can not find the requested wavelength available in its outgoing port, it will fail the reservation request and the connection request as well.

Several solutions and approaches have been proposed in the optical network domain to enhance the utilization and to reduce the blocking probability. Some of them were introduced in this thesis. Moreover, we highlighted the advantages and the disadvantages of the proposed solutions and we also provided major challenges and motivations yet to be improved. From there our contributions get started.

First, we proposed the Bottleneck Congestion Avoidance algorithm to enhance the use of label suggestion provided by RSVP-TE signaling for optical networks. This algorithm makes it possible for more efficient utilization of resources in optical networks by the decision that is taken by core nodes when they receive more than one request for the same wavelength and output port. As a result, the proposed algorithm improves the limitation of the suggested label approach that was described in Section 2.1.

Second, we proposed the Traffic Engineering Technique that enhances the network utilization and reduces the blocking probabilities. In addition, the proposed T-E technique has a cost efficient way to update the network nodes states to the T-E server. This is done without the need for the core nodes involvement. Moreover, the required memory for the proposed technique is sufficiently low. By building an optical network model, we demonstrated the proposed T-E technique efficiency by comparing the blocking probability of the proposed technique with the OSPF through simulations. We concluded that with the use of our proposed technique, the probability of utilizing the wavelengths in a bottleneck is clearly increasing and remaining within the range [MAR /LN*bottleneck NFP, 1] where bottleneck NFP is the Number of Fibers in the bottleneck node Pair. MAR is the Maximum Available Resources from ingress to egress of the worst route that the bottleneck can be part of; where the worst route is the route that has the node pair with the minimum number of fibers.

This thesis proposed the Bottleneck Congestion Avoidance algorithm. Future work is needed to simulate this proposed algorithm in real-time manner and to compare it with the traditional suggestion label that used in RSVP-TE to evaluate the algorithm performance. In future work, we would also like to apply the PSLS algorithm on RSVP-TE to suggest wavelengths in its path messages, and so to be able to distribute the network resources in advance when the connection requests initiated. Moreover, future work would also be needed to consider and simulate the proposed algorithms and techniques in bidirectional LSP setup case.

REFERENCES

[1] S. Xu, L. Li, and S. Wang, "Dynamic Routing and Assignment of Wavelength Algorithm in Multifiber Wavelength Division Multiplexing Networks," *IEEE JSAC*, vol. 18, no. 10, Oct. 2000, pp. 2130–37.

[2] J. Spath, "Resource Allocation for Dynamic Routing in WDM networks," *Proc. SPIE All-Opt. Net. 1999: Architecture, Control, and Management Issues*, Boston, MA, Sept. 1999, pp. 235–46.

[3] S. Subramaniam and R. Barry, "Wavelength Assignment in Fixed Routing WDM Networks," *Proc. IEEE ICC '97*, June 1997, pp. 406–15.

[4] I. Chlamtac, A. Farago, and T. Zhang, "Lightpath (Wavelength) Routing in Large WDM Networks," *IEEE JSAC*, vol. 14, no. 5, June 1996, pp. 909–13.

[5] Muralikrishna Gandluru, "Optical Networking and Dense Wavelength Division Multiplexing (DWDM)", Ohio State University online paper at: http://www.cis.ohio-state.edu/~jain/cis788-99/ftp/dwdm/index.html, November, 1999.

[6] Ayan Banerjee, et al., "Generalized Multiprotocol Label Switching: An Overview of Routing and Management Enhancements", IEEE Communication Magazine, Jan 2001, pp. 144-150.

[7] E. Rosen, et al., "Multiprotocol Label Switching Architecture", IETF RFC3031, January 2001.

[8] P. -H. Ho and H. T. Mouftah, "Capacity-Balanced Alternate Routing for MPLS Traffic Engineering," *Proc. IEEE 2002 Int'l. Symp. Comp. Commun.*, Taormina, Italy, July 2002, pp. 927–32.

[9] D. Awduche and Y. Rekhter, “Multiprotocol Lambda Switching: Combining MPLS Traffic Engineering Control with Optical Cross-connects,” *IEEE Commun. Mag. vol. 39*, Mar. 2001, pp. 111 – 116.

[10] Awduche, D., Malcolm, J., Agogbua, J., O'Dell, M. and J. McManus, "Requirements for Traffic Engineering Over MPLS", RFC 2702, September 1999.

[11] J.P. Lang et al., “Link Management Protocol”, Internet draft, draft-jetf-mpls-lmp-02.txt, Sep 2001.

[12] Andersson, L., Doolan, P., Feldman, N., Fredette, A. and B. Thomas, "LDP Specification", RFC 3036, January 2001.

[13] Ayan Banerjee, et al, “Generalized Multiprotocol Label Switching: An Overview of Signaling Enhancements and Recovery Techniques”, IEEE Communication Magazine, July 2001, pp. 144-151.

[14] L. Berger, "Generalized MPLS-Signaling Functional Description", IETF RFC 3471, January 2003.

[15] Braden, R. (Ed.), Zhang, L., Berson, S., Herzog, S. and S. Jamin, "Resource ReserVation Protocol -- Version 1 Functional Specification", RFC 2205, September 1997.

[16] Berger, L., Gan, D., Swallow, G., Pan, P., Tommasi, F. and S. Molendini, "RSVP Refresh Overhead Reduction Extensions", RFC 2961, April 2001.

[17] Ashwood-Smith, P. and L. Berger, "Generalized Multi-Protocol Label Switching (GMPLS) Signaling - Constraint-based Routed Label Distribution Protocol (CR-LDP) Extensions", RFC 3472, January 2003.

[18] Jamoussi, B., Andersson, L., Callon, R., Dantu, R., Wu, L., Doolan, P., Worster, T., Feldman, N., Fredette, A., Girish, M., Gray, E., Heinanen, J., Kilty, T. and A. Malis, "Constraint-Based LSP Setup using LDP", RFC 3212, January 2002.

[19] L. Berger, "Generalized MPLS-Signaling -RSVP-TE Extensions", IETF RFC 3473, January 2003.

[20] Kompella, K. and Y. Rekhter, "Signalling Unnumbered Links in Resource Reservation Protocol – Traffic Engineering (RSVP-TE)", RFC 3477, January 2003.

[21] Awduche, D., Berger, L., Gan, D., Li, T., Srinivasan, V. and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels", RFC 3209, December 2001.

[22] Adrian Farrel, "MPLS in Optical Networks", Version 2: October 2001 Data Connection Limited.

[23] Antonio Manzalini, et al, "Architecture and Functional Requirements of Control Planes for Automatic Switched Optical Networks: Experience of the IST Project LION", IEEE Communication Magazine, Nov 2002, pp. 60-65.

[24] Pin-Han Ho, Husssein T. Mouftah, "A novel Distribute Control Protocol in Dynamic Wavelength-Routed Optical Networks", IEEE Communications Magazine. November 2002, pp. 38-45.

[25] S.K. Lee, D. Griffith, V. Coussot and D. Su, "Explicit Routing With QoS constraints in IP over WDM", IEE Proc. Commun., Vol 149, No. 2, April 2002, pp. 83-91.

[26] Apostolopoulos, G., et al: "QoS routing mechanisms and OSPF extensions", RFC 2676, Aug, 1999.

[27] J. Moy, "OSPF Version 2," RFC 2328, April 1998.

[28] Qais MDM Al Rousan, Anjali Agarwal "Bottleneck Congestion Avoidance in GMPLS Networks" International Conference on Computer, Communication and Control Technologies: CCCT '03, July 31, August 1-2, 2003, pp. 22-26.

[29] K. Kompella *et al.*, "OSPF Extensions in Support of Generalized MPLS," Internet draft, draft-ietf-ccamp-ospf-gmpls-extensions-12.txt, October 2003.