NTNU

Norwegian University of
Science and Technology

Control plane of an OpMiGua network

Simulation or analytic study, analysing the feasibility of suggested solutions.

Mohammad Yaser Rahmati

Master of Science in Communication Technology
Submission date: July 2008
Supervisor: Steinar Bjørnstad, ITEM
Co-supervisor: Norvald Stol, ITEM

Norwegian University of Science and Technology
Department of Telematics

Problem Description

The main objective is to find solutions that can aid in implementation of an OpMiGua node using existing frameworks for e.g. IP and MPLS networks. A deep understanding of control plane mechanisms should be achieved during the study. Then, solutions including existing and new mechanisms for supporting the hybrid architecture of an OpMigua network should be sought. The master project involves an analysis of control plane mechanisms as well as a simulation or analytic study, analysing the feasibility of suggested solutions.

Assignment given: 20. February 2008
Supervisor: Steinar Bjørnstad, ITEM

NORGES TEKNISK-NATURVITNESKAPLIGE UNIVERSITET
FAKULTET FOR INFORMASJONSTEKNOLOGI, MATEMATIKK OG ELEKTRONIKK

MASTER’S THESIS

Student’s name: Mohammad Yaser Rahmati

Title: Control plane of an OpMiGua network

The main objective is to find solutions that can aid in implementation of an OpMiGua node using existing frameworks for e.g. IP and MPLS networks. A deep understanding of control plane mechanisms should be achieved during the study. Then, solutions including existing and new mechanisms for supporting the hybrid architecture of an OpMigua network should be sought. The master project involves an analysis of control plane mechanisms as well as a simulation or analytic study, analysing the feasibility of suggested solutions.

Start data:	20.02.2008
Deadline:	16.07.2008
Submission date:	16.07.2008
Department:	Department of Telematics
Supervisor:	*Norval Stol*

Trondheim, 12.09.2007
Norvald Stol

Preface

This master's thesis was introduced at the Norwegian University of Science and Technology (NTNU), department of Telematics in 2008. This is the completion of my fifth year of the Master program within the Telematics.

The main objective and activity in this thesis was to simulate the suggested solution for a control plan for OpMiGua. And this report contains the structure and result of the simulation which I made during the time of master's thesis.

I want to thank my advisors and associate professors Norval Stol and Steinar Bjørnstad who helped and guided my during the study of this thesis.

Tønsberg, 07 2008

Mohammad Yaser Rahmti

Contents

List of figures

List of tables

Summery

This report contains the introduction and result of the master's thesis studied in the last semester of my study in the NTNU in 2008.
The thesis was about studying a control plan for OpMiGua by using the existing technologies and solutions. For this propose a solution was suggested and a simulation is developed to test this solution.

The report starts with an introduction of different networking techniques and the motivation for the OpMiGua.
The report contains mainly three parts. The first part is the theoretical and technical background, where the Optical network and different technologies around that is introduced, including OpMiGua network and Control plan techniques as MPLS and GMPLS. The first part is from chapter 1 to 4.
Second part is about the use of GMPLS in the OpMiGua and the way of use it with respect to the classification of traffic. It is introduced in the chapter 5. Introducing continues with the solutions as GST_LSP, a path for the GST packets and how it can be established and how the packets can be labeled. It also introduces the SM_LSP, a path for SM packets and its properties and differences in compare to GST_LSPs.

The last part begin from chapter 6 through 9 where an analyses of the GST_LSP establishment is performed, a simulation structure is explained and the result of the simulation is discussed with respect to updating rate of the Gasco (Gaps statistic container) which is a database in the suggested solution. The simulation is developed in view of the establishment process of GST-LSPs and the kind of information about each GST_LPSs which is aimed to update the Gasco with. The simulation also extended to study the SM-LPS switching request (or request for establishment) by using the data in the Gasco, and see what is the degree of successfulness of the SM_LSPs requests.

The simulation of GST_LSP is about how this kind of path routes and establishes by using of GMPLS properties. This include finding shortest path in the studied network[1], establishing the path by using wavelength labeling introduced in the GMPLS and when a path is established , traffics generates and sends through the path.
The second process in the simulation is about what and how properties of the data which are flowing in the GSL path is going to be recorded and sends to the Gasco. The main property which is suggested to be importance in this study is peak delay or in another word, the longest burst duration in a path in different periods. This peak delay registers in the Gasco and is used to find the proper path for SM-LSP with respect to longest time a SM packet should with before it sends through the SM- path.

Chapter 9 which contain a conclusion and a further work ends the report.

[1] a network model is suggested and is used in the simulation

1.0 Introduction

Because of the ever increasing size of data and its associated overhead, a networking capability that enables transfer of a large and massive data set is needed in order to speed up the transmission rate and decrease delays. Non-optical networks have limited capabilities and smaller bandwidth because of the types of wires that are in use. But optical networks are constructed mainly of optical fiber and optical switches and hold a potential to distribute data at a large scale and high speed.

There are several optical networks introduced already. These are e.g. SONET (*Synchronous Optical NETwork*), OPS network (*optical packet switching)*, ORION network (*Overspill Routing in Optical Networks*), etc. SONET is a physical layer network technology designed to carry large volumes of traffic over relatively long distances on fiber optic cabling, SONET supports 51.84 Mbps. Higher levels of SONET signaling increase the bandwidth in successive multiples of four, up to approximately 40 Gbps.

Next, ORION is a hybrid network concept which combines (optical) wavelength and (electronic) packet switching, so as to obtain the individual advantages of both switching paradigms [1].

One another optical network is introduced recently which names OpMiGua (Optical Migration Capable Networks with Service Guarantees). This network aimed to fully utilize the potential capacity in optical fibers by sending different class of data packets in circuit switching and packet switching manners. Hybrid architectures were identified as attractive trade-offs for unbalanced traffics. A brief introduction of OpMiGua is given further in this thesis report.

To provide these two data switching method in OpMiGua in a cost-effective manner a set of control-plane protocols are necessary. These protocols are already defined in MPLS and GMPLS to enable signaling process and dynamic sharing of bandwidth [2]. GMPLS is an improvement of MPLS. MPLS switches and forwards data packets by numerical labeling. It means that the MPLS uses a number as a label, but in addition to packet labeling (numerical labeling) GMPLS also uses a series of other fractures to label data packets. These are fiber, wavelength, waveband and timeslot labels. A description of these is given on [3] and [4]. An introduction of (G) MPLS functionality is given later in this report.

The packet labeling and wavelength labeling ability of GMPLS can be suggested to switch data packets with low priority and data packets with high priority and guaranty in OpMiGua respectively. A related work was done by me and in that I studied a method to capable the OpMiGua with GMPLS control plan. This thesis will focused on the proposed solution method with respect to finding out how effective and reasonable the signaling time, frequency and speed of information distribution are for establishing an SM- path.

2.0 Optical network

Optical fiber refers to a technology where the Information transfers as a pulse of light which irradiates with a light source, through a fiber made of glass.

2.1 Optical fiber characteristics

Fiber is essentially a thin filament of glass that acts as a waveguide [5]. This glass is surrounded by a coating or cladding which acts as a shield and has a lower refractive index, preventing the light from passing across the fiber wire and reflecting it to the core. Fiber has a very low rate of data loss since the total internal reflection phenomenon is valid in this material. There are two types of fibers, one is single mode fiber and the other is multimode fiber. The core (glass) diameter in single mode fiber is much thinner in compare to multimode one. Because of its thin core a light wave propagate straightforward without any reflection through fiber core. Figure 1 shows two modes of fiber.

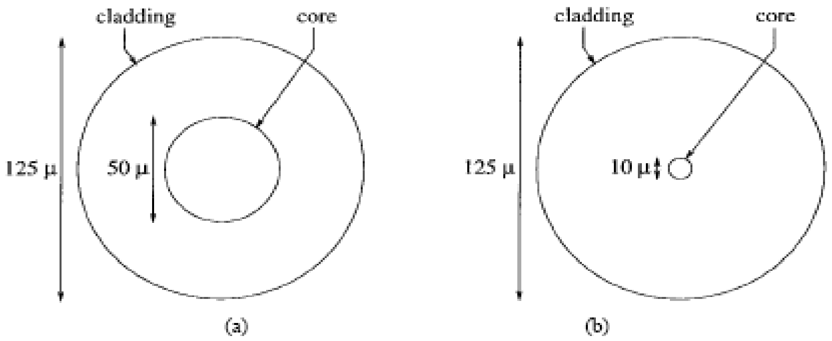


Figure 1: Compression of single and multi mode fibers [21].

As it is clear from figure 1 the diameter of the whole fiber cable is the same in both kinds of fiber modes. The difference lies on the size of core. This makes it possible that several light waves of with different wavelengths can be sent through fiber simultaneously.

Mode concept refers to the angle in which the light is send along the fiber.

Since a light is sent at different angles to the fiber in multimode fiber, it can experience some refraction which can reduce its speed. The light pulse spreads out in the time domain. In the worst case, this spreading causes an overlapping of light pulses. In such a situation it will be impossible to separate them and therefore it appears as an intermodal dispersion. This is a disadvantage with multimode fiber and can be solved or at least reduced by using grading-index fiber.

Multimode fiber core can be considered a series of layers with different refraction index. If there is a clear and sharp separation between refraction index layers, the fiber is called stepindexfiber. If the changes in index appear gradually the fiber is named gradedindexfiber. Figure 2 shows a gradedindexfiber. We can see that the region between core and cladding consists of gradual changing of glass which made the refraction index different in the core.

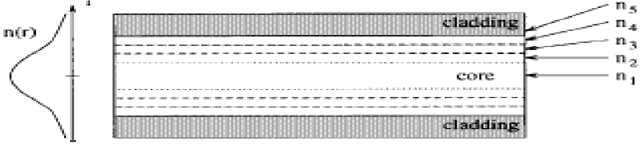


Figure2: Grading in a fiber[21].

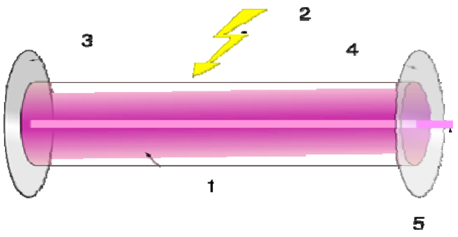
Gradedindexfiber reduces the angle degree which is needed to get total reflection, and can reduce the appearance of intermodal dispersion. In addition, the light pulses which we want to transfer over a long distance will be sent at the highest speed therefore, the light pulse will not spread in time domain.

Another solution is to reduce the numbers of mode in fiber, which results in reduction of intermodal dispersion. Therefore light pulse in the single mode fiber does not experience intermodal dispersion. In a single mode fiber the light does not have any refraction and hence no intermodal dispersion. Single mode fiber is mostly used for long distances since the light rays use almost the same time to propagate without any regenerating.

2.2 Light Generating and detecting

A fiber is equipped with a light source and a light detector. The light source sends a light pulse to the fiber and this pulse is detected by the light detector. Presence of a light pulse indicates a 1 bit and absence of light indicates a 0 bit. The signal converts to light pulse by the sender and this pulse is detected at the end of fiber by the detector.

The light source can be a laser, see figure 3. A laser is made of a slender lasing medium. There are two mirrors in the ends of the slender. When the lasing medium is excited by energy it emits light in all directions. The emitted light bounced between the two mirrors, results in amplification of the energy from the excitation mechanism in the form of light. One of the mirrors is transparent which allows some of the light to leave the laser and be sent through fiber. Light source can also be made of LED (Light Emitting Diodes). But the LED can just can be used for short distances and is used mostly with multimode fibers.



1: lasing medium, 2: energy (electrical power), 3:100 % mirror, 4: transparent mirror, 5: produced light pulse
Figure 3: Laser and its components

The light pulse is converted to the electrical signal by a detector. A detector is a photodiode. A photodiode has a PN transition layer. When the light pulse touches this layer, it will free an electron and therefore an electrical stream is begun to go from P layer to N layer. See following figure.

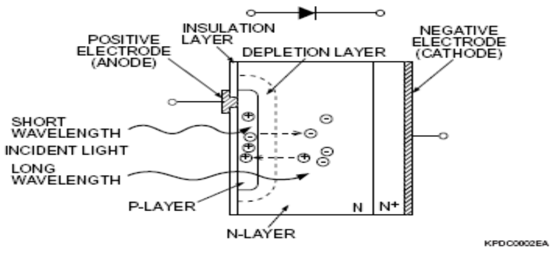


Figure 4: Producing of electrical stream in a photodiode

The optical fiber is immune against electromagnetic noise. This makes the bit error rate very small, around 10^-11. Fiber is flexible, has a high transmission capacity and speed, difficult to damage and easy to be used. They are cheaper than coacsial or twist pares cables.

2.3 WDM

A technique is needed to effectively realize the full capacity of optical fiber potentials. TDM (Time Division Multiplexing) was used to divide the bandwidth between several users by allocate the bandwidth in separate time slots. But due to limitations in TDM components it was not a very effective technique. Wavelength Division Multiplexing or WDM is a replacement technology to TDM. In WDM networks, the wavelength domain can be used to realize advanced and very efficient network functionality rather than only increasing the capacity carried in the fiber, as it is used today[8]. The WDM networks are equipped with switches which are opaque or transparent. In an opaque switch the optical signal carrying traffic undergoes Optical to Electronic to Optical (OEO) conversions but in a transparent switch the optical signal carrying traffic stays optical at all times from entry to exit [5].

WDM is a technology which multiplexes multiple optical carrier signals on a single optical fiber by using different wavelengths (colors) to carry different signals. This allows for a multiplication in capacity. By this all signals come to the receiver simultaneously and parallel. An illustration of this is showed below:

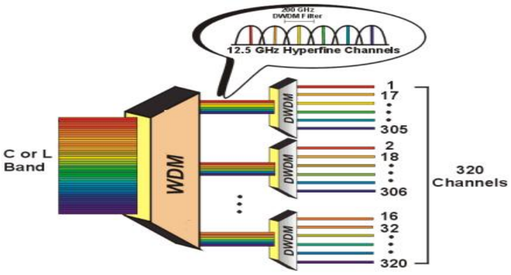


Figure 5: WDM of 320 wavelengths [20].

It shows the principle of multiplexing of 320 separate signals by MUX as a collected signal in sender node and it is demultiplexing to the separate signals in the destination node. The multiplexed signal sends the total optical power to each output port in the DEMUX, this requires that each receiver recover only one wavelength by using a tunable optical [20]. The filter's transmission peak is adjusted to a particular wavelength, so when the signal arrive to the filter, it blocks all other wavelengths accept what the filter adjusted to. (In WDM networks it is important to isolate each wavelength to avoid crystallization.)

2.4 DWDM

DWDM (Dense WDM) is an improvement of WDM which is able to send multiple wavelengths simultaneously. Thus a bigger portion of fiber capacity is used but it has a drawback in comparison to WDM. This is the crystallization that appears because of closed and dense channels. With a DWDM we can combine 160 to 320 wavelengths together with a speed of 10 to 40 Gbps per wavelength.

2.5 IP Over WDM

IP over WDM means the routing of IP packets directly over fiber. Network traffic has continued to increase with a staggering 115% per year on a global base [8]. With the networks which comprise of several protocols from IP layer to WDM layer we can not have an efficient usage of existing capacity in fiber due to slow processing of middle protocols in compare with increased traffic in IP layer and fiber capacity. This causes a bottleneck. In the traditional network architecture the ALL 5, ATM, and SONET/SDH layers are in between IP and WDM layers. Although the SONET has provided a guaranteed level of performance and reliability, to obtain the WDM gains, higher functionality and overall better performance in an IP network, a change of architecture to IP over WDM is needed. MPLS will be used as a form of traffic control for this new architecture. Figure 6 shows a schematic of two architectures.

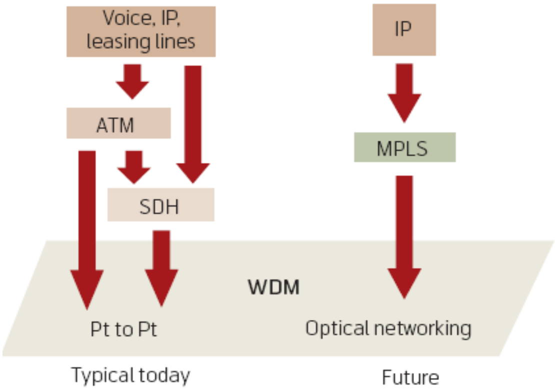


Figure 6: Simplification to IP over WDM [8]

ATM in the typical network architecture is responsible for traffic control and QoS-awareness.
Packets from an IP router come to an ATM switch. In an ATM switch, packets encode to cells of 53 bytes in length. 5 bytes belong to the header and the body consists of 48 bytes.
In the new architecture the functionally of traffic control is handed over to MPLS and the ATM layer is phased out.
SONET or SDH is responsible for data transmission and error protection mostly for voice traffic. Although the SONET is reliable, the bandwidth utilization and flexibility is quite bad.

So we see that using ATM and SONET in an IP to WDM network has more negative impact than what they offer. For example packets should encode to ATM cells which made overhead in the network, furthermore the ATM cells are fixed unlike IP MPLS labeling technique of variable packet lengths. SONET limits us to gain the capacity utilized by WDM technique. Phasing out these extra protocols of the network architecture brings a simple IP over WDM network. This makes the access network faster and the packets do not experience further encapsulation through the network. The IP routers get access to optical network without any mediator [8].

2.6 OCS over WDM

Use of WDM optical transport layer result in a transformation from ring topology based on SONET to a mesh topology using optical cross connections (OXC). In a mesh topology based on OXC traffic bypassing intermediate IP routers leads to fewer loads on routers .In meshed networks OXC are used to directly connect node pairs with high traffic between them. Related to this an optical core network can be designed to carry traffic all the way optically without any conversion to electrical signals and back to an optical signal. This kind of architecture is called a transparent switching network. Another design is to use some optical to electrical to optical (OEO) switches in the core network. In this architecture, optical signals undergo optical to electrical to optical conversions during switching. The former one is called opaque switching and is more reasonable today. Even though the applications driving the large scale deployment of transparent optical switches are not currently in place and the traffic demand does not currently justify the use of transparent switches that are cost effective at very high bit rates, it is possible that at some point in the future transparent switches may be deployed in the network [5]. By combining both transparent and opaque techniques, four different architectures can be achieved. These are represented in the following figure.

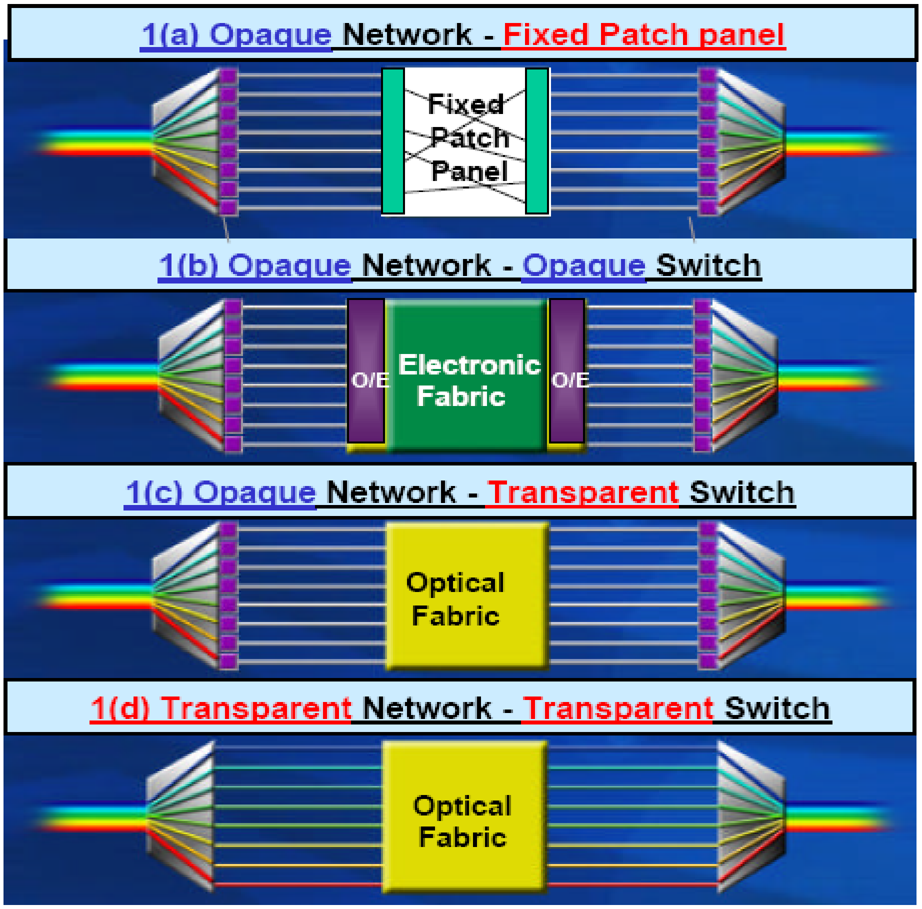

1(a) Opaque Network - Fixed Patch panel
Fixed Patch Panel
1(b) Opaque Network - Opaque Switch
O/E
Electronic Fabric
O/E
1(c) Opaque Network - Transparent Switch
Optical Fabric
1(d) Transparent Network - Transparent Switch
Optical Fabric

Figure 7: Node architecture for a core optical network [5]

Transparent network architecture may be a viable option for small-scale networks with pre-determined routes and limited numbers of nodes; it is not a practical solution for a core network. Due to the absence of wavelength conversion, physical impairments like dispersion, fiber non-linearities and polarization degradations appear [5], in such architecture opaque network architecture is preferred. Opaque switches will provide the grooming and multiplexing functions, as well as some of the necessary control and management functions, and will scale and decrease in cost with rapid progress in electronics.

2.7 Optical switching:

Mainly switching in an optical fashion can be done in three ways. Optical circuit switching, Optical packet switching and optical burst switching. They differ in way of resource reservation, application type and granularity.

2.7.1 Optical circuit switching:

In this technique a path is established and the necessary resource is reserved in advance before sending of data. The path is all-optical and the service offers in wavelength granularity. This kind of switching is also called wavelength routing. Circuit switching (CS) is often used for the application with loss guaranty and transmission with constant data rate. In the case of packet loss, the lost packet will be resent. The essential here in this kind of switching is that the setup process takes some time in *milli second or ms* scale and therefore the duration of connection should be in second or minuet scale to be a reasonable chose of switching. Bandwidth utilization is inefficient since the portion of capacity which is reserved for this switching is not fully utilized during data transmission. But the good thing with CS is the disciplinary and regularity of transmission where the packets sends in succeeding manner and also there is no buffering.

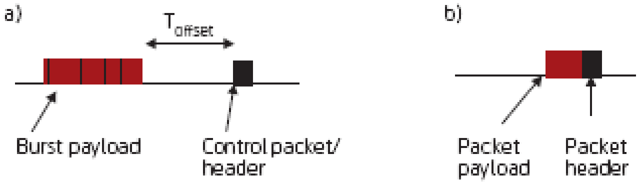
2.7.2 Optical packet switching:

In optical packet switching OPS, the data is sent in a packet level of granularity and there is no fixed path for this kind of transmission. Hence no resource reservation appears in advance. The packets will not arrive in a successive manner to their destination because each of them follows a different path through the network. Also some packets may be lost or buffered in the process. The packets consist of two parts namely header and payload. The payload will stay optical and unchanged during transmission, but the header may be processed electrically or optically. An OPS is based on the idea of separating forwarding from switching in the network nodes [8]. An OPS assumes in-band encoding of control information. Reading and reinsertion of packet headers with strict timing requirements are required due to short packet duration [8]. One of the advantages of OPC as compared to OCS is its efficiency resource utilization, since they introduced statistical multiplexing SM.

2.7.3 Optical burst switching:

Optical burst switching OBS is similar to OPS in many points, but the main difference between them is that the OBS consist of several packets' payload coupling together as a burst with a single header named burst control packet BCP. The header is always sending ahead and separate from the burst. By doing this the processing delay of header information will be reduced in comparison to OPS. In the packet switching each packet has a header to be processed, but here in OBS a single header is processed for several packets with the same destination and priority while other similar parameters are assembled in a burst. One of the similarities between OPS and OBS is that the header processing appears in the electrical domain.

The time between head arrival and burst arrival to a node is called offset and as a minimum it equals the time required to settle the switches and perform control processing in the node [8]. The following figure shows the main differences of OPS and OBS.



a) Illustrates that OBS typically has an "out-of-band" BCP, transmitted on a different wavelength and with a time-offset compared to the burst payload. b) illustrates that OPS typically has an "in-band" header, transmitted at the same wavelength and simultaneously as the packet payload [8].

Figure8: Main differences for transferring control information in OBS and OPS.

2.8 Hybrid switching:

In order to take advantage of both OCS and OPS/OBS functionalities in an efficient way, the combination of these two switching schemes is possible. So making a switching function based on two different switching techniques is called hybrid switching. Hybrid design opens several new degrees of freedom to bring packets and circuits closer together, optimizing the overall network design. There are already several hybrid networks with different focusing points, for example ORION consists of two switching modes. Overspill mode appears when a packet is detected as a packet of packet switching type and is sent to an electrical IP router or an OP switch. Normal mode appears when a packet is detected as a circuit switch type. The advantages of ORION are smaller OPS/OBS switches (less traffic seen, less processing) and a high utilization rate [6].

Another example of Hybrid architecture is OpMiGua which is aimed to fully utilize the total capacity in the optical network by sending some portions of traffic as guarantied traffic with high priority and the remaining traffic switches in a packet switch by using the remaining and underused capacity. These two kinds of traffic appear as OCS and SM. The next chapter introduces the OpMiGua concept and its architecture.

3.0 OpMiGua

Optical migration capable networks with service Guarantees or OpMiGua is an optical hybrid architecture worked by a cooperation of NTNU and Telenor. The aim of this architecture as its name describes is to achieve both service guaranty and resource utilization optimally. It is done by coupling two different switching techniques, OCS and OPS together. The OPS part can also be replaced with OBS. Traffic of OCS and OPS have different positive features which can be unified and give an optimal capacity utilization gain.

The OCS technique has a guaranty level of Qos where the data follows a fixed path without any packet loss. The drawback of OCS is its inefficient resource usage, which can be compensated by using the portions of underused capacity for sending OPS traffic which do not follow a fixed path or allocate recourse in advance. Thus providing guaranteed service in the strict sense and statistically multiplexed properties[7].

3.1 Classification of traffic

Based on OpMiGua properties there are two different traffics which will be processed and flowed through a OpMiGua node.

Traffics are classified based on their feature and importance and are as followed

1) GST traffic

2) SM traffic

The GST or Guaranteed Service Traffic follows the fixed CS path and has the high priority and Qos with no jitter.

SM traffic or Statistical Multiplexed traffic follows OPS manner and therefore the packets switch based on the information existing in the packet header.

The Qos for this kind of traffic is marked as Best Effort (BE) and has the possibility to be lost or buffered. They have fine granularity of resource utilization.

In order to find out how good the suggested properties is in practice, an experiment according to [9] shows that zero packet loss and no jitter for GST traffic, regardless of the load of SM traffic on the network. Having SM traffic increases lightpath utilization and it is demonstrated up to 98% utilization with an SM packet loss probability of less than 10^{-6} [9]. Next we will study how the coexistence of both SM and GST traffics happens in a hybrid node.

3.2 OpMiGua Node

OpMiGua node is mainly considered to consist of two parts parallel to each other. The main part is a WRON CS switch and the other part can be an OBS or OPS. In this description we consider OPS.

The optical circuit switching part is an OXC which forwards GST traffic. GST and SM packets are detected by a Polarization Beam Splitter PBS since they propagate in different polarizations in the physical layer. After splitting, GST and SM packets forward to OXC and OPS parts respectively. The GST packet goes through a Fiber Delay Line FDL before it reaches the OXC. The OXC can be static or dynamic and may contain wavelength converters at the output for reducing blocking probability [8]. If the OXC is static

it will be an S-WRON and will be very simple and reliable. If it is dynamic (D-WRON) and in such situation the connection can be established dynamically, and there is a need for a control plane. An illustration of hybrid node is as followed:

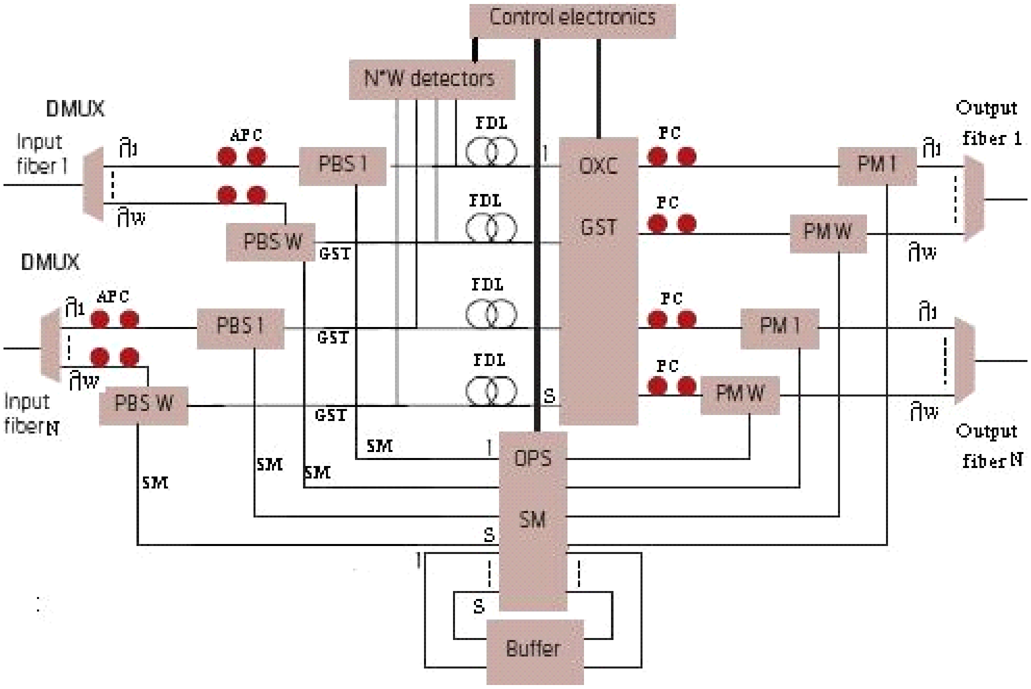


Figure 9: OpMiGua hybrid node

As the figure shows, when the GST packet is switching out of OXC it will go to the Polarization Maintaining coupler PM where it is a crossing between GST and SM packets. The control electronic device in the top of the switch controls the traffic flow and detects the absence or presence of GST packet and its utilization of capacity.

The SM switching part of OpMiGua can be an IP-router, Ethernet switch or OPS.
[8] introduce an architecture where the OSP is used for SM part. This is described in the following:

3.3 OPS part:

After the separation of SM packets from GST packets by Polarization Beam Splitters (PBS), this packet comes to a plane of the OPS part. The OPS switch is build of two successive switching parts called first switching and second switching. In the first switching; OPS consists of N vertical and separate planes. The number of planes is equal to the number of input fibers. Each plan has two tunable wavelength converters and an Array Waveguide Gratings (AWG) in between. The AWG is equipped by a buffer to save the packet in case of non-free AWG's output. The W outputs of each plan are connected to the corresponding input of the W separated and horizontal plan in the second switching part. A graphical presentation of this switch is shown in the figure 10.
Forwarding of SM packets undergoes two stages. In the first stage, the packet's wavelength at the output fiber is decided. In the second stage the packet is forwarded to the destined fiber. The packet forwarding process is as followed:
After the dimultiplexing of wavelengths to W corresponding fiber-lines which appears in the first phase of hybrid node, the MS lines are directed and fed to a converter. The converter is setting the wavelength so that the packet is forwarded to any of the desired outputs of the AWG. This makes it possible at the second stage, to freely choose at which AWG the packets will occur. This makes the first stage wide-sense non-blocking.
If the AWG's outputs are not free, the packet goes to the buffer existing in the AWG.
After passing the first AWG, the packet is sent through a converter before occurring at the input of the chosen AWG in the second stage. By setting the output wavelength of the converter, forwarding to any output fiber can be chosen. The packet then undergoes a third wavelength conversion, converting to a vacant wavelength at the output fiber before the

packet is sent through a passive coupler on to the output fiber [8].

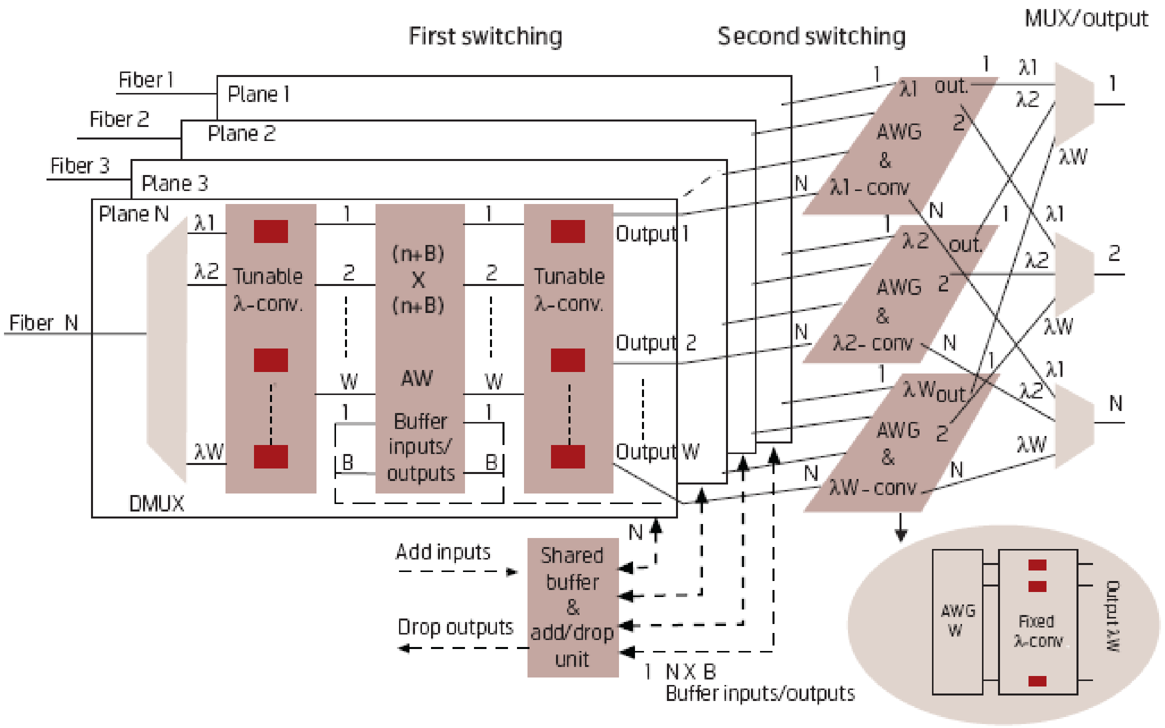


Figure 10: OPS node part design and architecture

It is valuable to notice that the DMUX is moved out of the plan in the hybrid node and placed before PBS. (Shared buffer does not exist in the hybrid case).

3.4 Source utilization:

During a traffic flow of GST type in a specific wavelength, the control device and detector device watch to detect if there is packet using the wavelength capacity in every time instance or not. Since the GST packet propagates though a wavelength as a light signal, so the detection of light by the detector unit means occupation of wavelength by that signal and the absence of light means free capacity as a gap. When there is free capacity, the control device forces the OPS unit to send an SM packet (if there is an SM packet waiting for transmission) further to the path by using that detected gap. In this way the hybrid switch sends both GST traffic and SM traffic in the same wavelength without any overlapping. This non-overlapping is because of the packets state of polarization through the network. By this policy SM packets can be sending over every wavelength which is underused for some time interval. In contrast to SM traffic, GST traffic sends over a dedicated wavelength in a fixed path. This static multiplexing mechanism in OpMiGua brings full utilization of capacity.

4.0 MPLS

The growth in both the number of users of the internet and their bandwidth requirements, make challenges for internet service providers ISPs to improve the switching performance, scalability and routing products. Since the access network was not able to work as fast as the growth of the Internet users and IP nodes in all dimensions, a switching mechanism was necessary. Parallel to the appearance of these problems and demands for a switching technique was worked out by the name of Asynchronous Transfer mode (ATM).
By this switching mechanism the requested performance could be achieved and soon it becomes clear that an important function of ATM is to forward the IP datagram. Hence this leads to the concept of IP over ATM. This network architecture places the ATM switches in the core network and connects the IP router (IP network) to it. See figure 11.

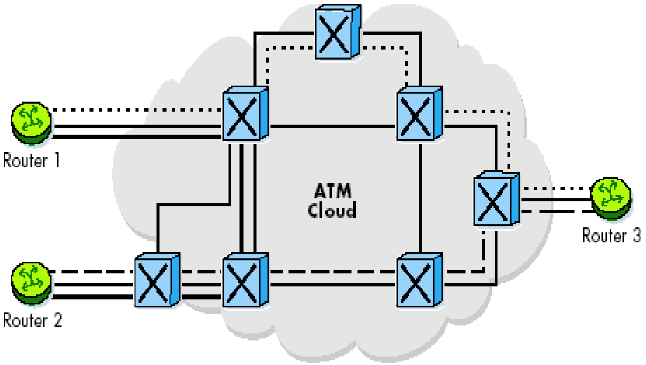


Figure 11: IP over ATM network

By doing things in this way, some integration of IP to the ATM was needed. ATM has an architecture model which differs a lot from IP architecture [10]. The Connection model and addressing scheme is completely different. To solve this problem some alternatives were suggested by different organizations. For example Cell Switching Router CSR of Toshiba, IP Switching from Ipsilon and Tag Switching from Cisco. All these had some similarities and differences and couldn't be published as standard switching protocols. Finally the Multiprotocol label switching was adopted by IETF based on IP switching and Tag switching [10]. The basic idea behind MPLS was the convergence of the flexibility and robustness of the IP control pane, and the simplicity and efficiency of the connection oriented forwarding mechanism in ATM. According to [11],MPLS integrates Layer 2 information about network links (bandwidth, latency, utilization) into Layer 3 (IP) within a particular autonomous system or ISP in order to simplify and improve IP-packet exchange. MPLS gives network operators a great deal of flexibility to divert and route traffic around link failures, congestion, and bottlenecks. It simplifies the routing processes in IP networks by using layer 2 labels for switching instead of hop-by-hop routing paradigms in the network layer.
MPLS speeds up network traffic flow and makes it easier to manage. MPLS involves setting up a specific path for a given sequence of packets, identified by a label put in each packet, thus saving the time needed for a router to look up the address to the next node to forward the packet to. It works in the layer 2 unlike the IP protocol which belongs to the network layer. In order to setup a path, MPLS also uses other signaling and routing protocols like Resource Reservation Protocol RSVP, Open Shortest Path First Protocol OSPF. MPLS is an algorithm that moves traffic away from the shortest path selected by the IGP onto label switched paths (LSPs). MPLS brings connection-oriented forwarding techniques together with the Internet's routing protocols by establishing a virtual connection between two points on an IP network.

The simplicity and flexibility of an IP network remain intact, while the ATM-like advantage of a connection-oriented network is exploited (12).

4.1 Terminology in MPLS:

To understand better how the MPLS works, the concepts which is considered in MPLS network architecture is introduced as follows:

1. Label
2. FEC: Forwarding Equivalent class
3. LER: Label Edge Router
4. LSR: Label Switched Router
5. LSP: Label Switched Path

Label in early MPLS is an entity with fixed length. Each data packet will be allotting a number as a label in MPLS switches in order to be forwarded along a path. This label contains no information about network address or destination address. Label can be swapped in every node that packet enters.

Forwarding Equivalent class is a partitioning mechanism where some packets with similar characteristics considered as a set or class. This is referred as Forwarding Equivalence class. For example, the equivalency can be based on the destination address where several packets with the same destination are assumed as a class of packets. Or similarity can be Qos requirements, forwarding types (multicast or unicast).

One important characteristic of FEC is the forwarding granularity. For example an FEC could include all the packets whose network layer destination address matches a particular address prefix. This type of classification gives a coarse forwarding granularity. Or a FEC can consist of packets that belong to a particular application running between a pair of computers. These packets are equivalent because they all have the same source and destination addresses or the same transport layer port number. This is a fine forwarding granularity [10].

Labe Edge Router is the ingress node in an MPLS capable network which attaches a label to a packet. Generally in MPLS forwarding mechanism, the IP address checks just one time and it happens in LER where the destination address checks and the packet allots a label.

Label Switched Router is an intermediate node which forwards the packets based on the label and incoming interface value. LSR decides the outgoing interface and attached the packet with a new label. The change of label refers as label swapping and appears based on the information stored in a Label information base LIB in the LSR.

Label Switched Path is the path which establishes along the LSRs for Data traffic of a FEC. The packet forwarding in a LSP can have different labels in different part of the LSP because of the swapping mechanism in each LSR. In circuit switching the LSP is fixed during connection but if the LSP is done for a packet switching, it can be varied during the connection.

To shows the relationship between these entities in a MPLS-base network a data flowing case is described in the following.

Figure 12 illustrates a situation where two LSPs are going to be establishing through a network. There are three hosts, A, B and C. Host A is a source node, B and C are destination nodes. Node 1 is an LER, node 4 and 5 are Egress LSRs and the other nodes act as intermediate LSRs. When packets generated form host A with destination address arrives to the LER, it indicates the packet's FEC and starts to establish a path through the network to the destination host.

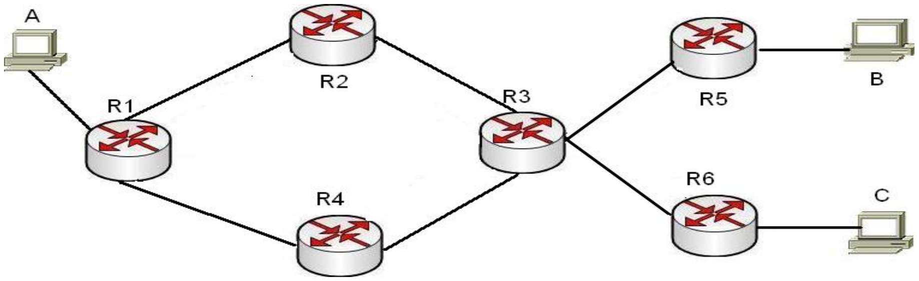


Figure 12: A case model for MPLS

The first phase is Label distribution is performed by Resource Reservation Protocol RSVP. This is a signaling protocol. To make the RSVP protocol compatible with MPLS, some new feathers are added to the RSVP, referred to as RSVP-TE. One of these feathers is LABEL_REQUEST and LABEL object added to the PATH and Resv message respectively in RSVP.

After identifying the FEC of the packets, the LER sends a PATH message towards the Egress LSR. The Egress node sends a Resv message in return following the same way as the Path message. When the Resv message reaches to each intermediate LSR , it will update the in coming and outgoing label columns in the LIB by label numbers and reserve the requested resources. When the Resv message reaches the LER node, the LSP is actually established.

Consider that after the signaling (label distribution) phase for each LSP the LIB in each LSR is updated as the following:

LIB 2

Interface In	Label In	Destination	Interface Out	Label Out
LSR1	12	B	LSR 3	5

LIB 4

Interface In	Label In	Destination	Interface Out	Label Out
LSR1	15	C	LSR 3	8

LIB 3:

Interface In	Label In	Destination	Interface Out	Label Out
LSR2	5	B	LSR 5	19
LSR4	8	C	LSR 6	25

Tabel 1 : LIBs in different nodes

Interface In refers to the node which packet sends from.
Label In is the number which the incoming packet is labeling with.
Interface out refers to the outgoing node.
Label out is the label which is swept to be the node that LIB belongs to.

During connection time the packets destined to node B labels by number 12 in the LER node (ingress node) and forwards to the LSR 2 where the label swapping performs and packets sends further by label 5 toward LSR 3. LSR3 swaps label 5 by label 19 and send it to LSR5. The label 19 swaps again in LSR5 to label 7 and forwards to egress (LSR 5). Egress node removes the label 7 and sends the packet to the Host B.

A similar process is performed for traffic destined to C. and the traffic in the two LSPs flow as illustrated in figure 13. Paths for traffic destined to B and C are respectively referred to as LSP1 and LSP2 in the following figure.

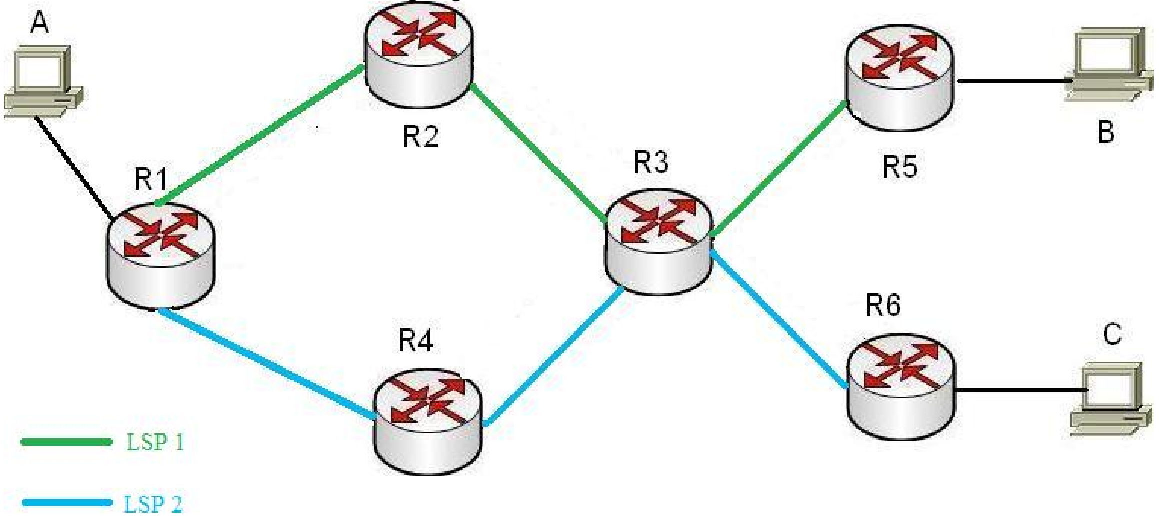


Figure 13: Two LSP paths based on MPLS Principle

To switch and forward traffic in this way we get Traffic Engineering (TE) requirements and better utilization of resources.

4.2 Traffic Engineering in MPLS

TE is the mechanism for mapping customer data flows onto an existing physical topology. It implements a process for routing data through the network according to a view of resource availability and the current and expected traffic volume. The class of service (CoS) and Qos that the data requires could also be factored into this process. TE helps the network operator make the best use of available resources by spreading the traffic load over the physical links and allowing some links to be reserved for certain classes of traffic or for particular customers [12].

With MPLS traffic engineering, we do not have to manually configure the network devices to set up explicit routes. Instead, we can rely on the MPLS traffic engineering functionality to understand the backbone topology and the automated signaling process.

MPLS traffic engineering accounts for link bandwidth and for the size of the traffic flow when determining explicit routes across the backbone.

The need for dynamic adaptation is also necessary. MPLS traffic engineering has a dynamic adaptation mechanism that provides a full solution to traffic engineering a backbone. This mechanism enables the backbone to be resilient to failures, even if many primary paths are recalculated off-line [13].

We can summarize some aspects of TE as follow:

- TE is concerned with performance optimization of operational networks.
- Traffic performance: A major goal of Internet TE is to facilitate efficient and reliable network operations while simultaneously optimizing network resource utilization and traffic performance.

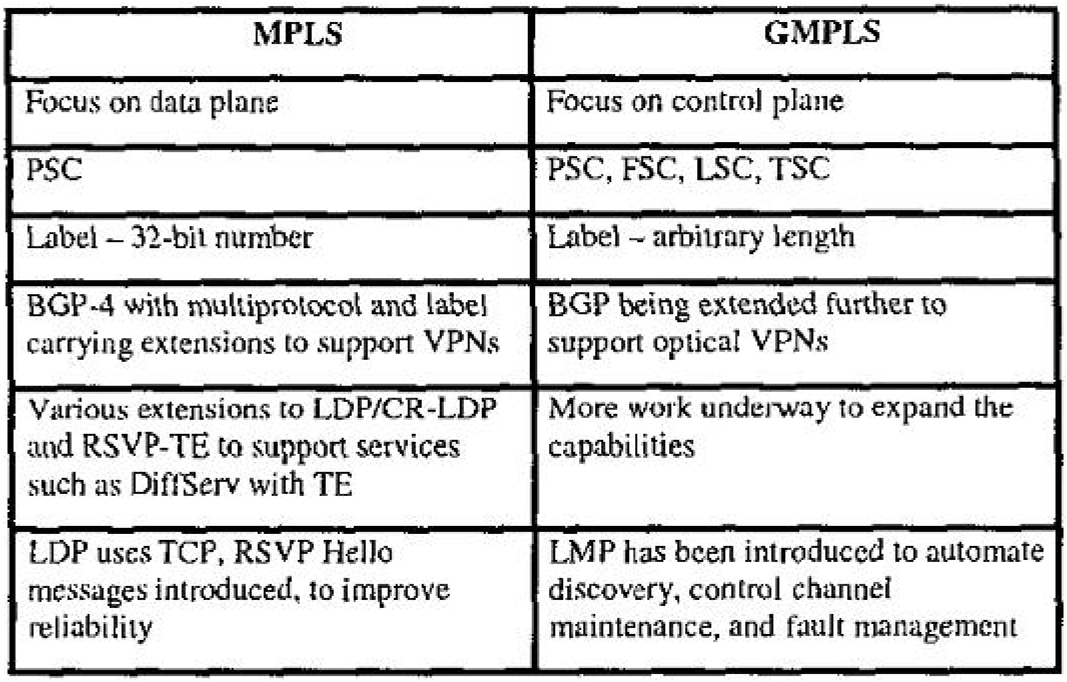
- Qos: Traffic oriented performance objectives include the aspects that enhance the Qos of traffic streams.
- Resource utilization: Resource oriented performance objectives include the aspects pertaining to the optimization of resource utilization.

To equip the MPLS with these fetchers the TE properties are added to some protocols of the network layer. These protocols are signaling the RSVP protocol and routing the OSPF protocol which are extended with the aim of the TE properties and referred to as RSVP-TE and OSPF-TE. An explanation of these protocols is given in the next section.

4.3 GMPLS VS MPLS:

MPLS in the first step was worked out basically for solving the performance and routing problems for the IP networks and ATM. The labeling in this solution was just based on packet labeling (fixed length number), but later the idea of using MPLS for other networks e.g. Optical networks, wireless networks, etc extended the MPLS to GMPLS where G stand for Global and states the ability of MPLS to be used in other networks. GMPLS brings the possibility to establish an LSP by using other feathers in various networks as labels. Labeling in GMPLS can be performed by using whole fiber, waveband, wavelength and timeslot in addition to packet labeling.

One of the main differences between the original MPLS and GMPLS is their functional focus. The original MPLS mainly focuses on the data plane. On the other hand, GMPLS focuses on the control plane that performs connection management for the data plane for both PSC interfaces and now packet switched interfaces. Another difference between MPLS and GMPLS is that the original MPLS requires the LSP to be set up between routers at both ends, while GMPLS extends the concept of LSP set up beyond routers. The LSP in GMPLS can be set up between any similar types of LSRs at both ends [14].



MPLS	GMPLS
Focus on data plane	Focus on control plane
PSC	PSC, FSC, LSC, TSC
Label – 32-bit number	Label – arbitrary length
BGP-4 with multiprotocol and label carrying extensions to support VPNs	BGP being extended further to support optical VPNs
Various extensions to LDP/CR-LDP and RSVP-TE to support services such as DiffServ with TE	More work underway to expand the capabilities
LDP uses TCP, RSVP Hello messages introduced, to improve reliability	LMP has been introduced to automate discovery, control channel maintenance, and fault management

Table 2: Differences between MPLS and GMPLS approaches [14].

By using different labeling mechanisms defined in GMPLS the switching can appear in different levels as follows [14]:

4.3.1 PSC - Packet Switch Capable

PSC interface can switch the received data on a packet-by packet basis. This interface recognizes packet boundaries and can forward data based on the content of the packet header. The label carried in the shim header is used in this interface. All kinds of labels used in PSC interface are defined as MPLS label.

4.3.2TSC - TDM Switch Capable

TSC interface forwards data based on the data's time slot in a repeating cycle. This interface can multiplex or demultiplex channels within a frame such as the SDH payload.

4.3.3 LSC - Lambda Switch Capable

LSC interface forwards data based on the wavelength on which the data is received. Therefore, this interface can recognize and switch individual lambdas within the interface.

4.3.4 FSC - Fiber Switch Capable

FSC interface forward data based on a position of the data in the real world physical spaces. Therefore, this interface can switch the entire contents to another interface (without distinguishing lambdas, channels or packets). Fiber switching system switches at the granularity of an entire interface, and cannot extract individual lambdas within the interface. This interface uses fiber label.

Some enhancements are required to (G) MPLS routing and signaling protocols to address the particular characteristics of optical transport networks. These enhancements summarize as follows:

- Enhancements to the RSVP-TE and CRLDP signaling protocols to allow the signaling and instantiation of optical channel trails in optical transport networks and other connection-oriented networking environments [15].

- Enhancements to OSPF and IS-IS interior gateway routing protocols (IGPs) to advertise availability of optical resources in the network (e.g., bandwidth on wavelengths, interface types) and other network attributes and constraints [15].

The enhancement and extension of RSVP to RSVP-TE involve the concepts of hierarchical LSP and Bidirectional LSP.

4.3.5 Hierarchical LSP:

Hierarchical LSP occurs when a new LSP is tunneled inside an existing higher-order LSP so that the preexisting LSP serves as a link along the path of the new LSP. In this section we illustrate how lower-order LSPs trigger the formation of higher-order LSPs. See figure 14. The ordering of LSPs is based on the link multiplexing capabilities of the nodes. Nodes at the border of two regions, with respect to multiplexing capabilities, are responsible for forming higher-order LSPs and aggregating lower-order LSPs [15].

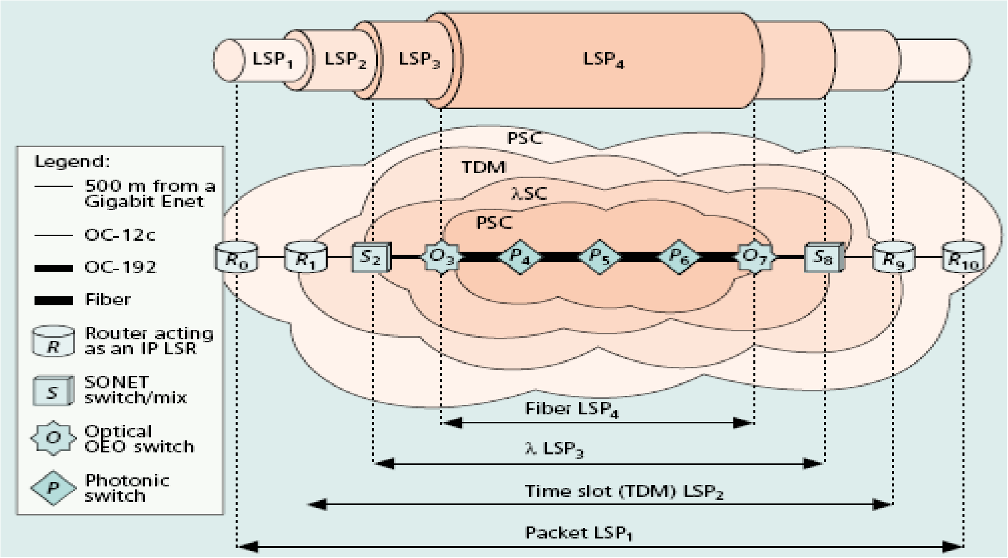


Figure14: Hierarchical LSP

4.3.6 Bidirectional LSP:

In bidirectional the LSP is assumed that both directions of such LSPs have the same traffic engineering requirements, including fate sharing, protection and restoration, and resource requirements (e.g., latency and jitter). The term *initiator* is used to refer to a node that starts the establishment of an LSP, and *terminator* is used to refer to the LSP destination node. Note that for a bidirectional LSP, there is only one initiator and one terminator. Additional methods have been defined to allow bidirectional LSPs' downstream and upstream data paths to be established using a single set of Path/Request and Resv/Mapping messages. This reduces the setup latency to essentially one initiator-terminator round-trip time plus processing time, and limits the control overhead to the same number of messages as a unidirectional LSP [15].

In addition to signaling protocol extension, the routing protocols as OSPF are also extended to make it compatible with GMPLS. With extensions OSPF to OSPF-TE, engineering (TE) attributes will allow nodes to exchange information about optical network topology, resource availability, and even policy information. This is done via properly defined *link state advertisements* **(LSAs)** that are maintained in a database. Constraint-based optical path computation, a special case of a routing and wavelength assignment (RWA) algorithm, is then used to select the lightpath's subject to the specified resource and/or policy constraints. The constraint-based optical path computation algorithm makes use of the topology and resource information stored in a database [16].

5.0 GMPLS in OpMiGua

With respect to capabilities in the GMPLS and OpMiGua a suggested solution to the control plan for OpMiGua is presented and studied. See appendix A.
Since the OpMiGua handle two types of traffics, one with guaranty Qos and resource access and one with best effort BE Qos and fine granularity of resource utilization, establishing of paths for these two types of traffic is appearing in two different ways according to appendix A. So we have two different LSP in this control plan represented as GST-LSP and SM-LSP respectively for GST traffic and SM traffic.

5.1 GST-LSP:

This path is established through OpMiGua in a wavelength level. The label to be used is a wavelength assigned in the Ingress node and has the possibility to be swapped in each intermediate LSR. A fixed path is established by signaling protocol RSVP_TE based on the computation of the shortest path using information distributed by the OSPF-TE protocol. GST-LSP actually follows the standard LSP setup pattern introduced in GMPLS.

5.2 SM-LSP:

The establishment of a path for SM traffic is dependent on some parameters as available gaps in a particular GST_LSP which overlaps the path for a particular SM traffic. Since the SM traffic forwards by a BE Qos level we cannot reserve a fixed and predefined portion of capacity for this kind of traffic. To set up a path for SM packets we suggested establishing the path using the packet labeling capability in GMPLS in the signaling process.
By this each SM packet labels by a number in the ingress node and eventually swaps in each intermediate LSR. Labeling SM traffic by the use of packet labeling will satisfy the full capacity utilization in OpMiGua. The calculation of the path is actually based on information about underused capacity in a GST-LSP, which appears as gaps. The use of the OSPF_TE protocol to distribute this kind of information is both time consuming and causes overhead of resource utilization. Another method is to choose a path randomly and assume that there will be underused capacity in the chosen path. This is in contrast with OpMiGua proposes because it is possible that the chosen path has not sufficient capacity and the packet should wait or drop while it is possible that there are some better choice of SM-path. So to achieve the necessary information for path calculation a more centric solution is suggested. In this solution the information about gaps or underused capacity periodically sends to a database during a GST-LSP connection time. So when a SM-LSP path wants to be calculated the information requests from the database. This database refers as Gasco (Gap Statistics Container). The GST-LSP path with most gaps will be used to forward the packet labeled SM traffic. It should be mentioned that the SM-LSP is not fixed and can cross and establishes through several GST-LSPs.
The Gasco model has two messages in order to exchange information. These messages are Update and Request used for sending gaps statistics to Gasco and getting these statistics from Gasco respectively.
For more information about the OpMiGua control plan and these two LSPs read appendix A.
Establishment of a LSP: As I explained the routing process for each traffic type is different, the GST traffic routing is based on the standard protocol OSPF-TE and SM traffic routing according to the suggested solution which performs by checking the Gasco.
But the establishment process is the same for both traffic types. The RSVP-TE is used for this purpose. How the RSVP_TE works for OpMiGua is explained in the oncoming section.

5.3 RSVP-TE: This protocol has a set of messages which is used for different purposes. These messages are described in [17] and [18]:

Path: This message is used to send the reservation request to the next node (LSR) on the path returned by the OSPF-TE. The Path format contains parameters such as *Session Attribute*, *Label request*, *Sender descriptor* and etc.

Resv: This message is a response to the Path and acknowledges the success of the reservation. The Resv format contains parameters such as *Session*, *Resv Confirm*, *Scope*, *Notify Request* and etc.

PathTear: This message sends from the sender when the connection is going to be finished. This is sending to release the reserved resources and tear down the path. The parameters in this message are *Session*, *RSVP_HOP* and *Sender descriptor*.

ResvTear: In order to acknowledge the release of resources, this message sends to the sender node which has sent the Path Tear. It has the same parameters as Resv message except the *Resv Confirm* parameter.

PathErr: This message sends in case of for example, link failure. Each PathErr message carries enough information to identify the RSVP session that triggered the message. If this is a transit LSR, it simply forwards the message. If this LSR is the ingress router (for this RSVP session), it has the complete list of all nodes and links the session should traverse. Coupled with the originating node information, the link can be uniquely identified.

ResvErr: This message indicates that the Reservation request was rejected by admission control due to unavailable resources.

Hello: This message sends periodically to confirm that the path is alive

Notify: in case of link failure this message notified the nearest node.

The two important messages in this set are the Path and Resv which I will illustrate and show how they perform in a sequential manner as described in [19]:

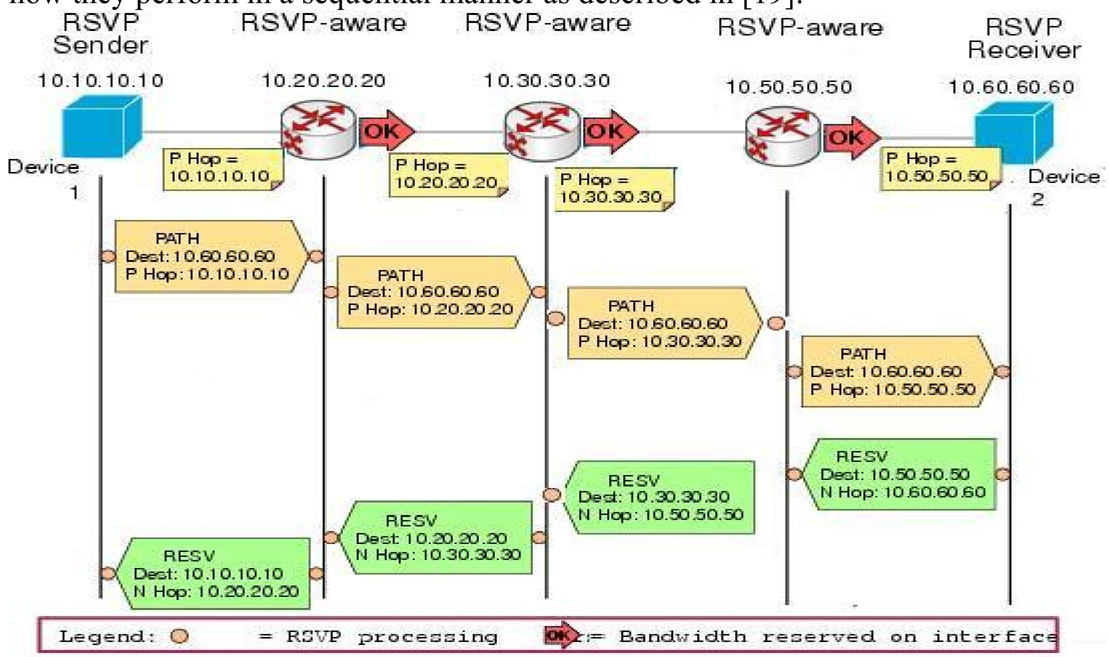

RSVP Sender
10.10.10.10
Device 1
RSVP-aware
10.20.20.20
RSVP-aware
10.30.30.30
RSVP-aware
10.50.50.50
RSVP Receiver
10.60.60.60
Device 2
OK
OK
OK
P Hop = 10.10.10.10
P Hop = 10.20.20.20
P Hop = 10.30.30.30
P Hop = 10.50.50.50
PATH Dest: 10.60.60.60 P Hop: 10.10.10.10
PATH Dest: 10.60.60.60 P Hop: 10.20.20.20
PATH Dest: 10.60.60.60 P Hop: 10.30.30.30
PATH Dest: 10.60.60.60 P Hop: 10.50.50.50
RESV Dest: 10.50.50.50 N Hop: 10.60.60.60
RESV Dest: 10.30.30.30 N Hop: 10.50.50.50
RESV Dest: 10.20.20.20 N Hop: 10.30.30.30
RESV Dest: 10.10.10.10 N Hop: 10.20.20.20
Legend: ○ = RSVP processing
OK := Bandwidth reserved on interface

Figure15: Establishment process through the network

1. The Device 1 which is the ingress node originates an RSVP message called Path, which is sent to the same destination address as the data flow for which a reservation is requested (that is, 10.60.60.60). One of the parameters contains in the Path message a label number which can be the same, or can be changed by the interceptor node. In case of GST-LSP establishment The Path message sends through the wavelength which is thought to be reserved for the GST-LSP.

2. The Path message is intercepted by the CPU of the RSVP-aware switch identified as 10.20.20.20 in figure 15, which sends it to the RSVP process. RSVP creates a path state for this data flow, storing the values of the session, label, sender Tspec, and P Hop objects contained in the Path message. Then it forwards the message downstream, after having replaced the P Hop value with the address of its outgoing interface (10.20.20.20 in this example).

3. Similarly, the Path message is intercepted by the CPU of the following RSVP-aware switch, identified as 10.30.30.30 in Figure 15. After creating the path state and changing the P Hop value to 10.30.30.30, this switch also forwards the message downstream.

4. The Path message gets to the RSVP-aware switch identified as 10.50.50.50, which processes the message, creates the corresponding path state, and forwards the message downstream. Notice that the P Hop recorded by this switch still contains the address of the last RSVP-aware switch along the network path, or 10.30.30.30 in this example.

6. The RSVP Receiver at Device 2 (egress node) receives the Path message with a P Hop value of 10.50.50.50, and it can now initiate the actual reservation by originating a message called Resv. For this reason, RSVP is known as a receiver-initiated protocol. The Resv message carries the reservation request hop-by-hop from the receiver to the sender, along the reverse paths of the data flow for the session. At each hop, the destination address of the Resv message is the address of the previous-hop node, obtained from the path state. Hence, in this case Device 2 sends the Resv message with a destination address of 10.50.50.50. The Resv message contains, among other things, the following objects: for example the label number reserved by egress node.

7. When the RSVP-aware router 10.50.50.50 receives the Resv message for this data flow, it matches it against the path state information using the received session object, and it verifies if the reservation request can be accepted based on the following criteria:

– Policy control: Is this user and/or application allowed to make this reservation request?

– Admission control: Are there enough bandwidth resources available on the relevant outgoing interface to accommodate this reservation request? Is there any wavelength available? And if yes, the label allocating occurs.

8. In this case, we assume that both policy and admission control are successful on 10.50.50.50, which means that the bandwidth provided by the Tspec in the path state for this session is reserved on the outgoing interface (in the same direction as the data flow, that is from Device 1 to Device 2), and a corresponding "reservation state" is created. Now switch 10.50.50.50 can send a Resv message upstream by sending it as a unicast packet to the destination address stored in the P Hop for this session, which was 10.30.30.30. The N Hop object is also updated with the value of 10.50.50.50.

9. The RSVP-aware switch identified as 10.30.30.30 receives the Resv message and processes it according to the mechanisms described in steps 7 and 8. Assuming policy and admission control are successful also at this hop, the bandwidth is reserved on the outgoing interface and a Resv message is sent to the previous hop, or 10.20.20.20 in this example.

10. After a similar process within the switch identified as 10.20.20.20, the Resv finally reaches the RSVP sender, Device 1. This indicates to the requesting application that an end-to-end reservation has been established and that bandwidth has been set aside for this data flow in all RSVP-enabled switches across the network.

5.4 Gasco and TE:

Traffic Engineering TE mechanisms contribute to minimize network congestion and improve network performance. TE modifies routing patterns to provide efficient mapping of traffic streams to network resources. This efficient mapping can reduce the occurrence of congestion and improves service.

The Gasco solution also modify the routing of the SM-packet in such a way that it always computes a path by searching the best path just among the already established GST-LSPs which match with SM-traffic source and destination nodes.

So the Gasco has an overview of a reduced network topology. It means that just the nodes and links which are encountering with traffic flow, are stored in the Gasco. And the links which are 100 % free have no table in the Gasco.

Since the SM-packets have Best Effort priority, it will never be reserved a full free wavelength through the network for this kind of traffic. This constraint enforces the routing process to find the proper path amount GST-LSP which is alive and its state is periodically updated in the Gasco database.

The Gasco makes it possible to create a TE aware network. This is because some of the requirements of the TE such as network performance, efficient resource utilization, service improvement are achieved by Gasco.

It should be mentioned that due to the routing of SM-traffic one suggestion is the use of OSPF-TE which introduced the Constraint-based routing protocol CSPF. But calculating the best path should be based on the information such as gaps number, gaps length and other attributes. This information should be broadcasted and flooded by OSPF-TE protocol through the network for all nodes. This requires some modification of the protocol which is not appropriate and advantageous. In addition, flooding of information by OSPF-TE are time and resource consuming in comparison to Gasco which is a centric solution.

So to summarize the Gasco properties in relation to the Traffic engineering we list the following points:

1. Reduction of network topology
2. Overview of the underused portion of the network resource
3. Constraint-based routing instead of destination based routing

One contradiction between Gasco and TE is the Qos requirement in the TE.

It is important that the Gasco database is always active and accessible. This can be achieved by using of two or more interconnected Gasco databases depends on the network size and expansion. This will make the system more reliable and increase the Qos.

6.0 Information distribution through signaling

An optical network has the capability of huge traffic transition because of its large capacity and fast transition rate. The traffic propagates in an optical network by light radiation. The abilities of optical fiber causes that the traffic shape changes very quickly during a traffic flow through a data path, traffic shape depends also in application type. The information such as quantity of data, the used portion of resources by specific data traffic, remaining resource capacity and setup time of a LSP path are critical and important in order to utilize the network's available resources optimally.

The hybrid network, OpMiGua, aims to benefit the huge capacity enabled by optical fiber transmissions. This is done by sending GST packets through the OpMiGua network as absolute priority traffic and sending the SM packets using the remaining and unused capacity in a GST-LSP paths. A control plane for this network for being able to establish, maintain and tear down paths in the network for the different traffic classes is an issue. For this, we suggest a solution based on GMPLS technology, which was introduced in [4].

To find advantages and disadvantages with the proposed solution, some critical issues are considered in this study. We must pay special attention to the following issues:

1. Setup time for a GST-LSP path
2. Update time and frequency for updating the gap information in the database
3. Distribution of information about Gaps stored in the database, distribution time and frequency.
4. Database response time

The most significant issue is the update frequency of gaps that appears in a GSL-LSP path. This is important because the SM-LSP is calculated and is set up based on gap information stored in the database. Hence, the information must be frequently updated at all times

6.1 Analysis of setup time delay in OpMiGua

6.1.1 Signaling: The first phase in the setup of a GST LSP in OpMiGua is the signaling process, where the Resource Reservation Protocol (RSVP) is used. In addition to reservation of necessary resources, the signaling enables Traffic engineering in the network. This influence parameters like e.g. traffic performance, operational performance and Qos.
The setup time of GST LSP paths includes the following parameters:
Total delay= $\Delta_I + \Delta_p + \Delta_{pr} + \Delta_q$
Initiation delay: The time that the initiator node (source node) uses to initiate a request, given by processing time in the node, denoted as $\Delta_{I.}$
Propagation delay: The time that the signaling packet uses to traverse through the network. Light travels through vacuum at a speed of 3*10^8m/s. Light can also travel through any transparent material, but the speed of light will be slower in the material than in a vacuum depending on the *refractive index and the angle of* radiation[21]see next section. Propagation delay is shown as $\Delta_{p.}$

Processing delay: Is the time the node requires for checking the message and decides a proper action related to the message content. This delay depends on the node architecture and logic. The delay is $\Delta_{pr.}$

Queuing delay: In case of high load and contention, the time the signaling packet stays in a buffer, depends on the buffer length and output rate. Queuing delay denotes as $\Delta_{q.}$

To find the most significant part of the total delay, an analysis for each part will be done on the case described in [4]. The network in this case consists of six switches, two routers and asymmetric fiber links between switches. The optical network part of the case looks like a fish with a tail. The fish body has a ring topology consisting of four switches. Ring topology is among the simplest form of topologies, and is common in deployed networks. To find the propagation delay $\Delta_{p,}$ actual distances are needed. So each switch of the fish-body is placed in a city in the southern part of Norway to form a ring structure. The four cities are: Stavanger, Bergen, Trondheim and Oslo. The cities are arranged from node nr 1 to node nr 4 respectively. We assume that the fiber lines follow the roads or rail. Hence the distance represents the road distance between cities not air distance.
Figure 16 and figure 17 show the case and it's orientation on the map.

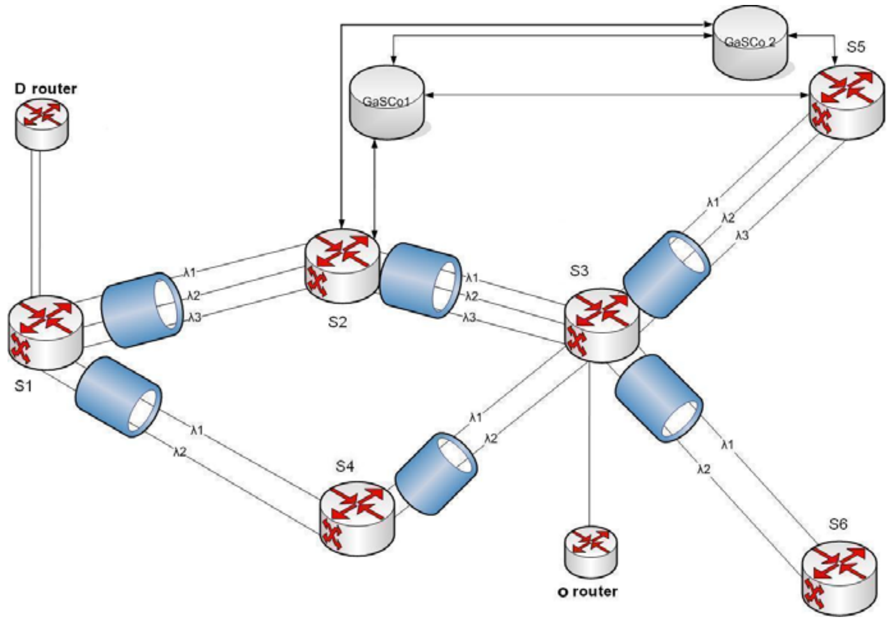


Figure 16: The case under study

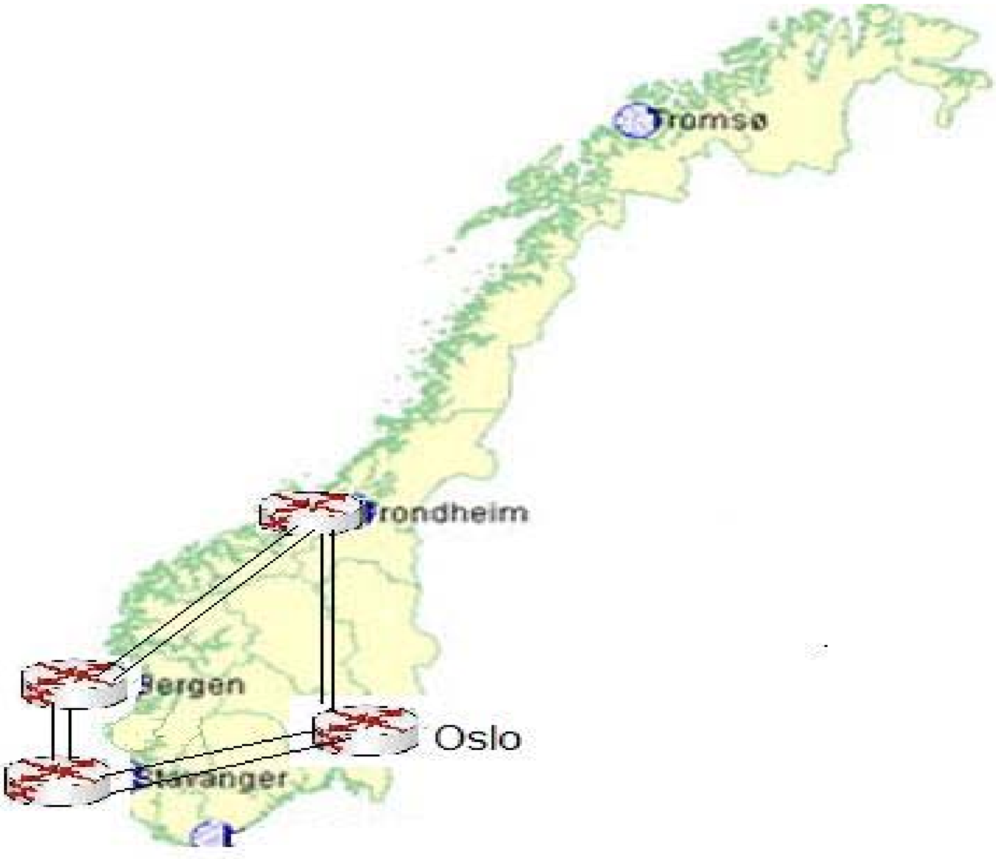


Figure17: The ring structure of the case oriented on a geographical map

6.1.2 GST LSP:

The GST traffic is the traffic with high priority, requiring high Qos, using according to OpMiGua definition, a fixed circuit path through the network. As an example, analyzing the setup delay of a LSP for GST traffic, using the MPLS signaling ability namely RSVP_TE, we consider a path from node nr 1 (Stavanger) via Oslo to node nr 4 (Trondheim). Beginning with propagation delay, it depends on the distance and speed of the packet transmission.

Δ_p = distance / speed

The distance between Stavanger and Oslo is 452 km and from Oslo to Trondheim is 500 km. The light speed in fiber depends on the fiber refraction index. The ratio of the light speed in a vacuum into the fiber is known as the fiber's *refractive index* (n). This is given by:

$n_{f =}\ v_c / v_f$

The light speed in fiber is:

$v_f = v_c / n_f = 3*10\^8 / 1.52 \approx 2* 10\^8$

So the propagation delay become as followed:

$\Delta_p = ((452*10\^3) / (2*10\^8)) + ((5*10\^5) / (2*10\^8))$

$\Delta_p = 4{,}8\text{ms}$

This is the time the PATH message of the RSVP-TE protocol requires for traversing to its destination node in Trondheim. The RESV message is traversing the same distance, but in the reverse direction; consuming the same time. Hence, the round trip time is:

$\Delta_p = 2* 4.8\text{ms}=9.6\text{ms}$

In addition to this, the propagation delay also depends on the sending speed.

Sending delay calculates based on packet size and total capacity in the fiber, which it goes through.

$\Delta_s = L/C$. Where L is the length of the packet and C is the capacity in fiber.

Since the signaling packet is an SM packet and an SM packet is almost 1500 Bytes [8], and a fiber has a capacity of 10Gbps, the sending delay is:

$\Delta_s = 1500*8/ 10\text{Gbps} = 1.2\mu\text{s}$

Another factor that impacts the setup time is the processing delay $\Delta_{pr.}$ Processing includes 1)Identifying the content of the received packet, 2)What to do with the received request,3) Is there enough resources available to handle resources as needed 4) formulate and dispatch message about discount or acknowledgement of required resources.

To approximate the delay, an experimental analysis is suggested. In our simulations we can send a packet through an OpMiGua switch. The processing delay in this case is the time interval from the instance at which a switch receives a PATH message up to the instance at which it sends out a PATH message further to the next node.

This kind of experiment requires laboratory facilities which are not available for this project. A similar experimental is already done by other researchers. This is reported in [22], and therefore the result of this can be used here. Since other kind of optical switches can be suitable for an OpMiGua network, the experimental result on an OXC switch is considered for this analysis. By this experiment they implement a GMPLS capable wide-area network. RSVP-TE protocol is used for signaling and measuring the setup time and processing delay in each optical switch across the path. In the processing delay, they have included the time a PATH message and a RESV message requires for being processed in the switch. This is precisely what we are looking for in this paper. Hence, we use these results as an approximation for our case. The result of this experiment [22] shows the processing delay as follows:

PATH-processing delay = 0.091119 s

RESV-processing delay =0.090852 s

So the processing delay is:

$\Delta_{pr=}$ 0.091119 +0.090852 =0.181971 s

This is a per switch processing delay.

Since RSVP is considered to be of SM_H type and according to the local policy and ratio change policy described in [4], this packet does not experience the buffering; therefore no buffering and queuing delay.

Buff-delay=0.

The total setup time is:

Total delay= Δ_p +Δ_{pr} = 9.6ms+0.181971 s

Analyzing the setup time shows that the most dominant part is the processing delay. If we consider a constant distance between nodes in a network with identical nodes (switches), the general setup time is

Total delay = N*(2Δ_p +Δ_{pr})

Where the N is the number of nodes a packet crosses on its way to the destination. 2Δ_p is the round trip time given from Δ_p both for the PATH message delay and for the RESV message delay.

6.2 Database updating frequency

Updating of Gasco is a critical and important issue. It is important that the database has the up to date and correct information about the traffic shape and available capacity in different GST_LSP paths, that are active in the network. The frequency of updating DB depends on the granularity of the information about the available capacity. It means that whether the information of unused capacity which sends to the DB should represent the result of measurement in a short time instance or a longer and fixed time instance. If we consider very frequently sending every gap detected by the control unit in the switch further to the DB (measurement over a short time), then the DB will be inconsistent and exposed to strain. In addition, this kind of updating takes more time in comparison to the traffic flow rate in the fiber. The GST type of traffic require a determined capacity as a resource to be allocated over a fixed path, so this property of GST-LSP allow us to consider a situation where the DB is updated over a longer period of time by information such as max capacity allocated by a specific GST-LSP, and remaining capacity in that wavelength. In this situation, the DB updates first when the GST_LSP path has been established and has begun to send the packets. During GST_LSP duration, the DB can be updating several times in order to have a better picture of how GST packets use the allocated capacity. The updating period can be predetermined for example every T seconds or it can update when the traffic shape is changed dramatically. So in this situation the updating frequency depends on the duration of the GST_LSP path and its way to use the allocated capacity. The following figure shows the max-allocated capacity and the use of that over a period.

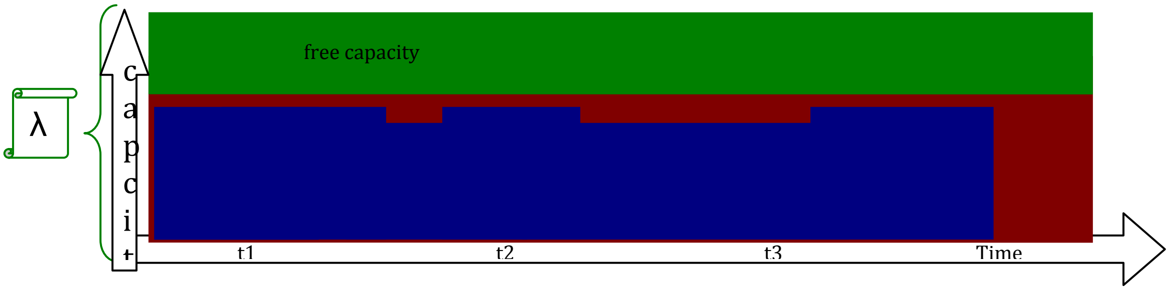


Figure18: Relation between allocated capacity, used capacity and free capacity in GST- LST

Refer to the figure 18, λ represent the wavelength total capacity, the red part indicates max-allocated capacity and the blue represents the actual usage of some portion of the capacity in different time instances.

Following this strategy makes it possible and gives the control unit a kind of ability to measures the traffic or inverse of traffic, namely gaps denoted by A in a period T.

Average amount traffic (or gaps) δ in time i is given by:

$\delta\ (i)$= A(i)/T

A: the carried traffic volume or inverse of traffic.

By repeating it several times and sending all of them to the DB, it will calculate an estimated traffic in that LSP in current time using this:

$$\delta\ (\text{current time}) = \sum_{i=0}^{k} a_i \delta\ (i)$$

Where Σ ai=1.

ai is a facture which indicates the impotence of ith measurement and its weight in the calculation of the estimated traffic in current time. The kth measurement is the most

significant one because it gives a value which represents the newest information about capacity usage in the GST path. And therefore the ak has the highest value:
ak>ak-1>ak-2>ak-3>ak-4>...>ak-i>ak-i-1>..>a1>a0
The period T is normally between the millisecond to second scale in most traffic analysis. Since the OpMiGua is an optical network the resource usage manner in this type of networks changes very quickly so the T should be shorter than what is normal in other traffic analysis. If we take the T in the second scale so the average traffic δ will be the same and it will not give a good estimation for the resource usage of a wavelength capacity in current time.
The T length is also depending on the intensity of inputting of packets in to the path, the length of packet and the sending speed of the fiber. Different applications have different and variable inputting intensity and packet length. If we consider an application which sends data with constant intensity and fixed packet length L we can calculate the using capacity for this kind of GST-LSP by multiplexing sending speed by intensity rate :
Total used capacity = (L\C)*µ
Where the C is the max allocated capacity of a wavelength for a GST-LSP and µ is the inputting intensity rate. This will give a very good picture of traffic shape and therefore the T can be very long.
The DB is updated just based on one parameter. This parameter is the estimated amount of traffic or gaps depends on whether we measure the traffic or gaps. It is because the traffic type of all packets is similar and there properties are almost identical. So we have a situation where the GST-LSP is setting up just for a determined application and traffic type. To make it clear we go back to our case example where there was established a GST-LSP path from Stavanger to Trondheim through Oslo. We can consider that this path is established to send the data of a video conference. This kind of application has a varied traffic flow and the intensity of inputting rate and packet length vary dramatically. Therefore, a very short T is required. In addition, the video conference is running mostly between 15 to 45 minutes; therefore the number of measurement K is also important in order to find out how the capacity is used and how much of the total capacity is unused. Measuring what happens in this GST-LSP path and updating the DB based on these measurements periodically during the connection time of this application (namely 15 to 45 minutes) also requires a database with huge memory capacity and high handle(or management) ability because of the huge amount of information sent to it in this period. Constricting the system and its logic based on this strategy arise these questions:
1. Is it feasible to have a DB with such huge amount of information representing estimation of just gaps volumes?
2. What is the purpose of this method?
If we have a situation where the traffic type (application type), its max capacity-requirements and max and min delay of its packets are predetermined, the measuring of gaps or traffic in a GST-LSP seems senseless.

To overcome these problems we should consider that the FEC in GMPLS for OpMiGua is not based on the traffic type but destination. If we classify the traffic by their destination node so the packets of different applications with same destination assumes to be in a same FEC. Hence a LSP path sends different traffic types simultaneously, a heterogeneous situation arises.

Consider if the GST-LSP path from Stavanger to Trondheim sends traffic of a different application like a video-stream, VoIP-traffic and file so there will be burst with different delays, bandwidth and other parameters.

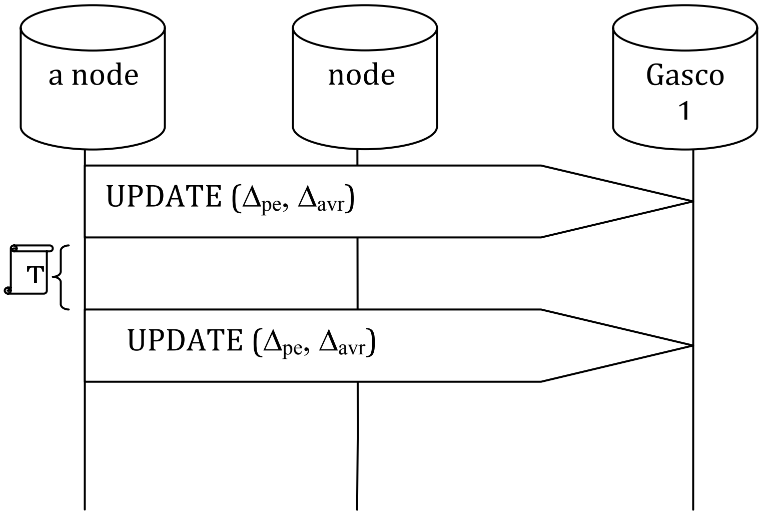
6.2.1 Parameters and model of measurement:

Changing the strategy from one dimensional measurement where it was concentrated on measuring the presentation of traffic or (absence of traffic) to a strategy where other parameters and properties of traffic are taken into account, makes the use of the Gasco more realistic and feasible.

Parameters:

In a heterogeneous situation where the traffic belongs to different applications, the parameters as maximum delay and average delay helps us to have an understanding of what happens in the GST-LSP.

Gasco will have overview of traffic with parameters as peak delay and average delay by updating:


a node
node
Gasco 1
UPDATE (Δ_{pe}, Δ_{avr})
T
UPDATE (Δ_{pe}, Δ_{avr})

Where Δ_{pe} refer to peak delay, Δ_{avr} refers to average delay.

Figure 19: Updating message with its parameter in each period of time T

The delay presents the time a SM packet should wait before it sends to the path.

To clarify how these parameters differ in various applications I take some applications and explain their properties and requirements. I remain that the packets aggregates and sends as burst through the network.

Burst:

A burst must be a sequence of consecutive received packets. The number of packets in a burst vary depends on application type.

For example a VOIP burst can have a maximum of 16 packets due to its low delay requirement. Or a burst of video services may have a size of 64 to 128 packets to send the data smoothly [22]. Packet bursting enables the transmission of multiple packets without the extra overhead.

Different applications have different restrictions and requirements and they classify as follows according to [22]:

- Data 0 (Voice). Highest priority, minimum delay.
- Data 1 (Video). High priority, minimum delay.
- Data 2 (Best Effort). Medium priority, medium throughput and delay. Most traditional IP data is of this kind.(SM type in OpMiGua)
- Data 3 (Background). Lowest priority, high throughput. Bulk data that requires maximum throughput and is not time-sensitive (for example FTP data) (SM type).

Bandwidth:

Bandwidth is often used as a synonym for *data transfer rate* - the amount of data that can be carried from one point to another in a given time period (usually a second). This kind of bandwidth is usually expressed in bits (of data) per second (*bps*).

In our case where a wavelength with bandwidth of minimum 2,5Gbps is shared between different applications, we can take the whole bandwidth commonly and calculate the burst duration of each application as follows:

Burst size/total BW= (number of packets in a burst size of a packet in byte *8)/ 2,5Gbps*

Consider a VoIP burst consist of 16 packets each with a size of 40 bytes.
VoIP burst:
16*40 *8 = 5120 bit
Duration: 5120b/ 2,5Gbps= 2,048µs

A video streaming's burst consisting of a minimum of 64 packets each with a size 120 bytes.
Video burst:
64 <B<128 p
120*64*8 =61440 bit
Duration: 61440bit/25Gbps= 24,5µs
24<Duration<49 µs

And a video conference burst consisting of 64 to 151 bytes.
Video conference burst:
64<B<1518 byte
1518*8 =12144 bit
Duration: 12144b/2,5Gbps= 4,85µs
0,2<Duration<4,85 µs

All numbers are according to the Cisco application presentation.

6.2.2 Traffic model:

The traffic granularity is considered to be burst as it is explained in the previous section. They are classified as four different kinds based on the application. Different bursts are distributed with different probability in the path. The GST burst distribution is such that the total traffic load is less than 100% in order to guaranty some underused capacity for utilization by SM packets.
The following table shows the traffic model:

plication	Traffic	Size (B)	Duration (µs)	Load %	Type
1	Burst	2560	2.048	30%	Voice
2	Burst	7680-15360	24-49	15%	Video
3	Burst	64-151	0.2-4.85	20%	Video- -con
4	Burst	30000-50000		15%	File
		10000[1]		80%	

B: byte

Table3: Data and properties of traffic model

A particular path of a GST-LSP establishes to send all or some of these applications (VoIP, Video Streaming and Video conferencing).
The load percentage shows also the importance of the application. For instance, the Voice traffic is real time traffic and should be send very quickly, therefore several independent telephone calls multiplex in the ingress node and send through the link simultaneously and de-multiplex in the egress node. This is the reason for the high load of this kind of application bursts.

In reality each of this application distributes differently, and sends by a distinctive distribution function, but I decided to assume a common distribution function, namely *negative exponential function* .see section 7.

[1] Main value is calculated as following:
0.8(2.5Gbps / (8*10000kb)) =2500burst/s*
Where 0.8 is the traffic load in the link and 80000 is the average bust length of burst in bit.

6.3.0 Analysis method:

There are various methods to analyze and dimension a system. According to [24] a system can be analyzed mathematically, by simulation or measuring.
Mathematical: a mathematical model develops based on a particular system which solves the problem analytically or numerically.

Measuring: a prototype is made and the behavior is observed as it appears in a realistic environment.

Simulation: a model is made of the studied system and its relevant event loops mimicked using a simulator.

In this work I use the simulation as a method to analyze the control plan assumed for OpMiGua.

6.3.1 The Simulation process

To model a simulation based on real system, there are some challenges to decide which aspects of the real system should be taken into account and which one should be eliminated. A real system has a lot of details which are perhaps not critical and do not impact the result of the simulation. It is therefore very important to have a good understanding of what happens in the real system and which details should be eliminated, to simplify the model and simultaneously decrease the affect of these eliminations.
In [24] a simulation process is drawn as follows:

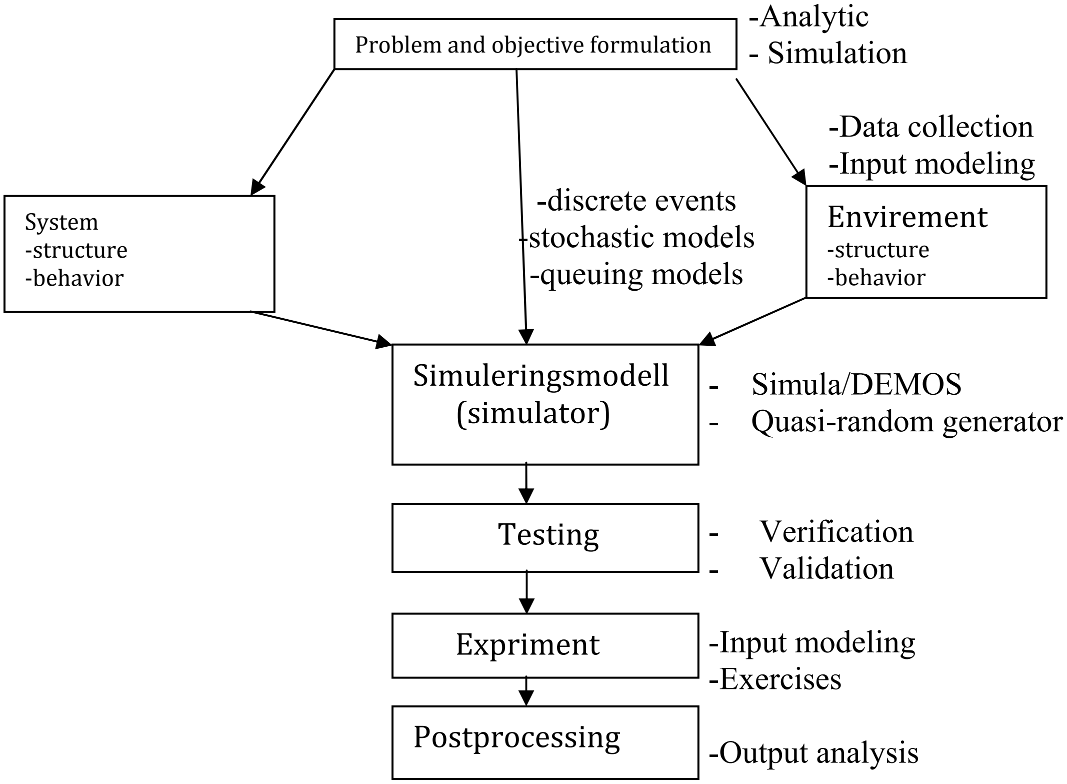


Figure 20: Simulation process

The simulation model can be developed based on the feathers and properties as *static vs. dynamic*, *discrete vs. continuous* and *deterministic vs. stochastic* according to [24].

Static: A static model is used to model a particular point in time.

Dynamic: A dynamic simulation model is used to represent a system as it changes over time.

Discrete: A discrete simulation model is used when the state of variables change only at a discrete set of points in time.

Continuous: The state of variables changes continuously over time.

Deterministic: When the simulation model contains no random variables.

Stochastic: When the simulation model contain random variables.

The model which I develop in this thesis is a dynamic, discrete and stochastic manner. It is because the system has parameters like LSPs which establishes dynamically and independent over the simulation time. The Discrete refers to the Gasco property where the parameters like peak delay, average delay, etc., updates in some point of times. And it is stochastic since it has some random variables like application type and burst size.

One of the challenges during this study was to decide in which base the Gasco parameters should be updated. Even though the parameter should be based on presentation or absence of packets properties flowing in a GST-LSP or based on application type properties.

7.0 Simulator model:

In this section a brief description of simulation model is given. See figure 21. This is based on the network model described in (4). It is supposed to model the simulation in two phases. The first phase represents the process of establishing a GST-LSP path and the other part represents the measurement and updating part after a successful establishment.

The important part is to simulate the updating of Gasco. When setup of a GST LSP is performed in simulation, the process of detecting of inputs traffic and its parameters began to run. It is important to know that this simulation does not contain any part of the SM LSP establishment process, but just the SM_LSP request. In the beginning I tried to simulate the gap measurement and study the amount of Gaps updated by Gasco versus the actual amount of gaps in a GST_LSP and course the updating frequency and traffic rate by this way. But soon I notice that the gap measurement did not give a good estimation of what happens in the path, because of the non sufficient amount of information about traffic flow and gaps. I changed the simulation model based on the second strategy mentioned in the previous section.

7.1 Phases in simulation model:

1. I consider a generator to generate *GST_LSP* requests . The generator becomes active when the simulation starts to run, and will generate LSP requests. Then the *GST_LSP*-request creates an object called OSPF to find the shortest path from ingress node to egress node. The OSPF object is terminated when it does its job, namely finding the shortest path for *GST_LSP*.

If the *OSPF* succeeds, the *GST_LSP* request creates a *RSVP* message and sends it to the first node in the path to reserve the necessary capacity.

The simulation also has six LSR objects corresponding to the six switching nodes in the network. The LSR objects are activated in parallel with the simulation and are kept alive for as long as the simulation is running. Their responsibility is to check and process the *RSVP* messages which are sent from previous nodes and are stored in its input-queue. The processing in the LSR is as followed:

1. Checking input-queue for RSVP message
2. Reserving the requested capacity (determine the lambda label)
3. Sending *RSVP* message further to the next node in the path.

In addition to these, the *LSR* also sends an *UPDATE* message to the Gasco object in the next phase. This message contains information about Path status with respect to capacity and also traffic shape in the *LSP* crossing this *LSR*. The update messages are sent periodically when the GST-LSP path is established and traffic begins to run. The Gasco object is responsible to collect and send in case of SM_LSP requests, the necessary info about underused capacity of a specified path. To achieve this ability, it creates messages as UPDATE and info-Request. Information stored in Gasco is used under an SM-LSP setup procedure.

A sketch of first phase of the simulation model is shown in figure 21. As the figure shows, the simulation model consists of six entities:

- Generator
- LSP-request
- OSPF
- Message
- LSR
- Gasco

The arrows between entities indicate the creation of entities and their relationship.

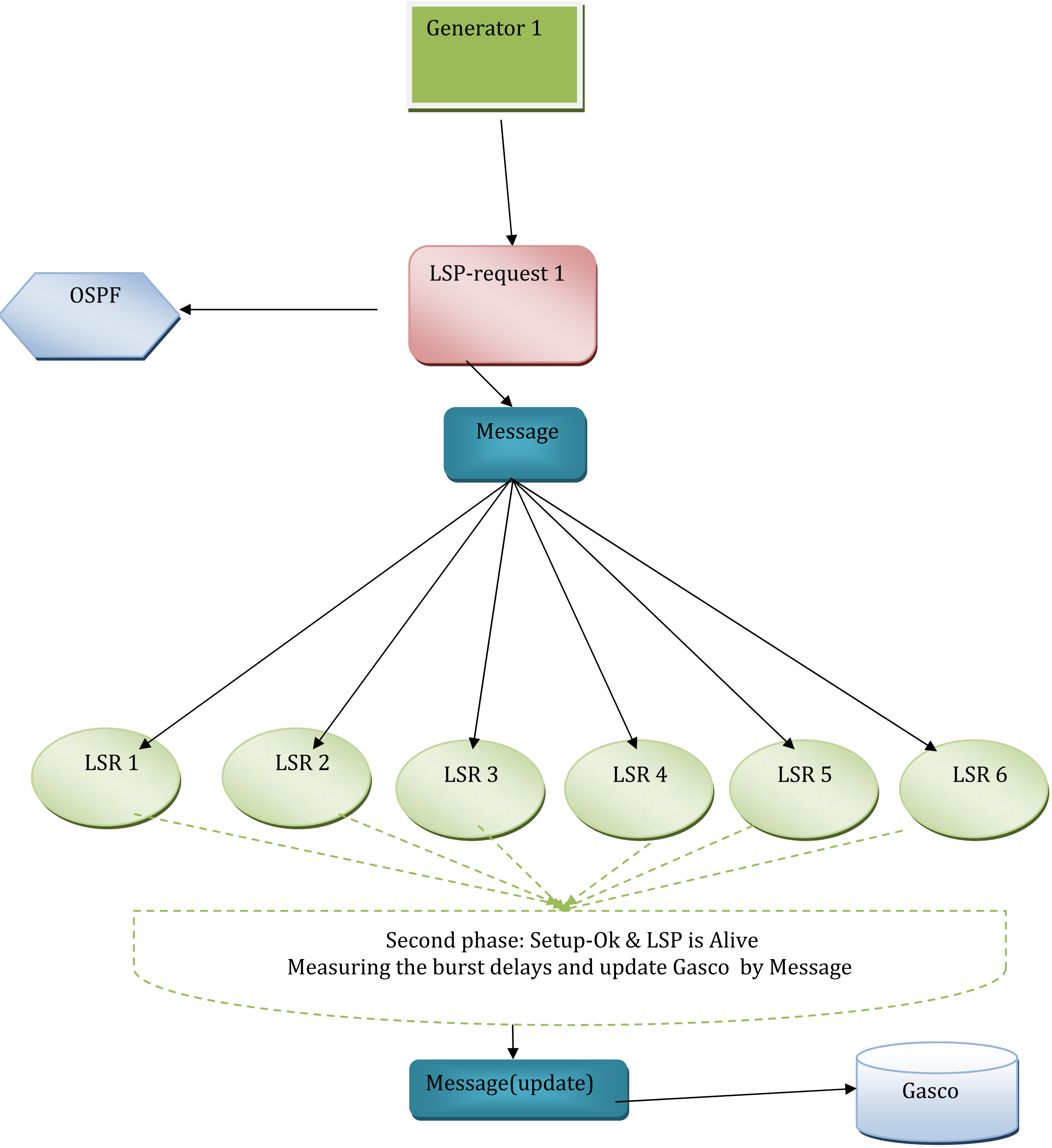

Generator 1
OSPF
LSP-request 1
Message
LSR 1
LSR 2
LSR 3
LSR 4
LSR 5
LSR 6
Second phase: Setup-Ok & LSP is Alive
Measuring the burst delays and update Gasco by Message
Message(update)
Gasco

Figure21: Simulation model

This simulation model illustrates the GST-LSPs setup processes.

Generator: Generating *GST_LSP*-request by some distribution function. The generator also draws up the application type set for every LSP simultaneously in a random manner. Distribution function can be a negative exponential function, gamma function or weibull

function. It also determines the ingress node and egress nod.
GST_LSP-request: This object is created and generated by the generator and has the ingress node and egress node as in-parameters. The *GST_LSP*-request will instantiate an *OSPF* object in turn and check to find out if the path return of OSPF has enough free capacity.
In the next phase *GST_LSP*-request instantiates a *PATH* message object to send through a dedicated path in order to reserve resource and setup path.

OSPF: This object returns the shortest path between ingress LSR and egress LSR given by *GST_LSP*-request object. OSPF object goes through the network table to check and find the shortest path. The network table shows the relation between LSRs. The network table can indicate if there is a link between two LSPs by showing the maximum capacity in this link or just shows zero if there is no link between them. See figure 22.

Message: *PATH* is a message created and distributed by *GST_LSP*-request. A *PATH* message is sent towards the first LSR in the path and after being processed by LSR it is sent further to the next LSR up to the egress LSR. The *PATH* message contains the information like requested capacity, incoming LSP id, label and etc.

LSR: this object represents a switch in the real network which runs all the time in the simulation life time. In this simulation model I consider an input-queue for this, where the incoming messages, like *PATH* message, is waiting there to be processed by the *LSR*. The *LSR* checks the input-queue and processes the message that arrive, when the processing is finished, it looks again at the queue for any new messages. See figure 23.
LSP has also a table that refers to LIB where it stores the information obtained during the processing of the message. The table will be updated by incoming LSP id, label type, outgoing LSP, etc.
Gasco: this entity consists of some arrays which are continually updated by the *UPDATE* message containing delay parameters. In this way the Gasco is updated with the newest traffic shape in each path. This estimation will be requested by the ingress node which wants to set up an SM-LSP.

	LSR1	*LSR2*	*LSR3*	*LSR4*	*LSR5*	*LSR6*
LSR1	*0*	*3* □	*0*	*2* □	*0*	*0*
LSR2	*3* □	*0*	*3* □	*0*	*0*	*0*
LSR3	*0*	*3* □	*0*	*2*□	*3* □	*2* □
LSR4	*2* □	*0*	*2* □	*0*	*0*	*0*
LSR5	*0*	*0*	*3* □	*0*	*0*	*0*
LSR6	*0*	*0*	*2* □	*0*	*0*	*0*

Figure 22: Network table

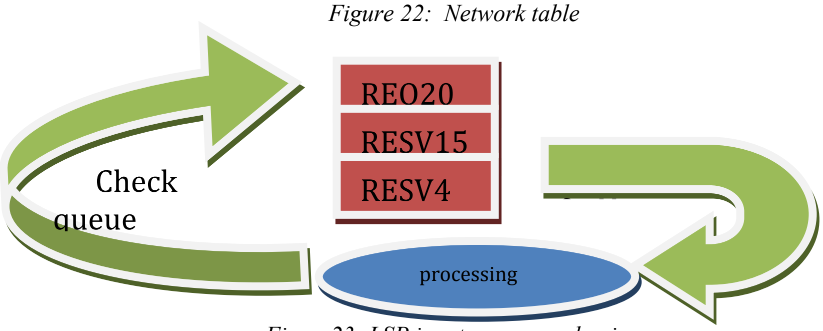


Figure23: LSP input-queue mechanism

2. In this phase when a GST-LSP is established and is alive a traffic object creates and sends to the alive GST-LSP in a randomly timed instance. Each object contains parameters such as burst delay and requested bandwidth.

So the GST_*LSP*- request entity or one of the *LSRs* entity which is in the path, register the parameters of all incoming objects over a period of time and sends the registered data further to the database by creating an *UPDATE* message directed to the *Gasco* entity.
A sketch of second phase of simulation model is shown in figure 24.

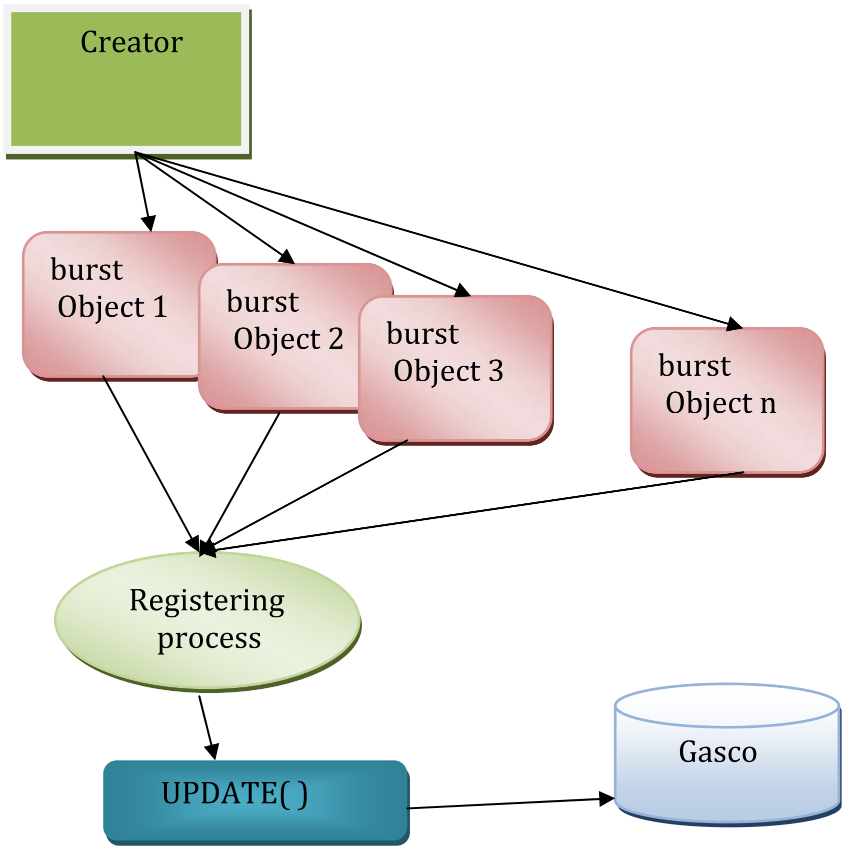


Figure 24: The process of instantiation of creator and inputting objects in the second phase during LSP alive period.

Creator: This entity instantiates when the establishment process with *PATH* message is successfully performed and a GST_LSP is established. It determines the application from the application set and based on this determination a burst with proper characteristics instantiates.

Burst Object: This object instantiates rapidly by the Creator and is sent further to the established GST-LSP. It contain parameters like burst delay, requested bandwidth, source node and destination node.

Update: When sufficient burst is sent to the GST-LSP an *Update* message instantiates to carry the information of the already flowed bursts to the *Gasco* table.

7.2 Statistics to be obtained:

Updating rate: Intensity or frequency in which the Gasco is going to be updated by.

Peak delay variation: Find out the variation interval of the max delay based on duration of bursts flow in a GST_LSP.

Accepted and rejected SM-LSP request: this is consider as something to be obtained if I can expand the simulation to simulate the SM-LSP as well.

8.0 The Implementation of Simulator

In this section I explain how I implemented the various entities of the simulation mode.
The following table shows the properties of and assumptions about some objects and entities:

Entities/	Dist.function	Main value	Comments
GST_LSP req	Deterministic	2 second	To establish max 12 GST path in the first half part of the simulation time
Burst	Negative exponential	2500bups[1]	A common function type for burst of all applications
SM_LSP req	Poisson	1.5 second	To have sufficient nr of request in the second part of simulation time

Table 4: Properties of different entities

The first thing that happens in the simulation is the instantiation of Generator. In theory I thought to let the generator generate LSP-request frequently, but during the simulation I deduced that this is very heavy for the DEMOS simulator.
What I did is to structure the generator to generate a restricted amount of numbers of GST-LSP requests. This is done by making a loop which runs as long as the simulation is running. The hold function in the loop has a determined holding time.
The applications set for a generated LSP-request is also determined randomly by the generator.
Since the process of Gasco updating is the most significant part, I suggest that when a LSP-request has been performed successfully, the GST-LSP will never tear down in the simulation time. Or in other words the simulation reflects the process of establishment and the life time of LSPs and what happens in that life time.

I defined the applications by numbering them and let the generator choose the set of the applications by returning a number that indicates the numbers of the applications. For example by returning a 3 it indicates that there are applications indicated by 3 and further in the set using this LSP.

LSP-request: This entity in reality is just a process consist of three phases. Start of each phase is totally depends on the result of previously phase in LSP-request.
After instantiation of the LSP-request with generator, the first phase is to start the OSPF entity object.
Second phase began after the result of OSPF; this phase is the PATH message instantiation.
The third phase is to fill the Gasco table with corresponding information about the GST-LSP properties and let the creator to create the burst and send them to the GST-LSP.

1. Main value is calculated as following:
0.8(2.5Gbps/(8*10000kb)) =2500burst/s 1burst per 0,0004 second*
Where 0.8 is the traffic load in the link and 80000 is the average bust length of burst in bit.

OSPF: as I described previously this object finds the shortest path for the requested LSP. It uses the queue mechanism to aggregate the generated LSP-requests and *coopt* them in turn. The core function of OSPF uses an algorithm which is as follows:

The neighbors of the ingress node are found from the network table, if one of the neighbors is the egress node, the process ends and a path is found. Else the neighbors of neighbors are found and checked against the destination node. This repeats until the destination node is found. If no nodes found as a source node after checking all possible ways, the OPSF terminates and returns a Boolean variable called "pathsfound" as false.
In case of a succeeded OSPF process the "pathsfound" is set to true. This variable is checked by LSP-request. If the result is true the establishment process is started, if the pathsfound is false the LSP-request terminates and rejects.

LSR: there are six objects of LSR entity. Each LSR has a dedicated in-queue. The LSR is implemented as a loop, thus it continually checks the incoming message.
The LSR also has LIBs which are Implemented as arrays to record the label, source node, destination node and next node.
These data is carried to the LSR by the PATH message.
At the beginning I implement the LSR in such a way that when it registers the information corresponding to the LIB, a new PATH message instantiates and sends it further to the next LSR, but it makes the simulation complex and unpreventable. I change the model to use just one PATH message to update the LIBs of all LSRs in the path. It makes the simulation easier and does not impact the effect of simulation in comparison to the real system. So by working based on this model the LSRs in-queue contains just the path message sent from the ingress node. According to the first modeling the in-queue of each LSR will contain path messages from both ingress node and previous node in the LSP path.

PATH: This instantiates with an LSP-request when the "pathsfoud" is true and sends to the in-queue of the egress LSR. PATH contains the parameters like ingress node, egress node, label, and intermediates nodes. The intermediate nodes are taken from the GST-LSP table updated by the OSPF with nodes found for the GST-LSP path.
A graphical comparison of the real path message sending and its design in the simulation is shown in the following.

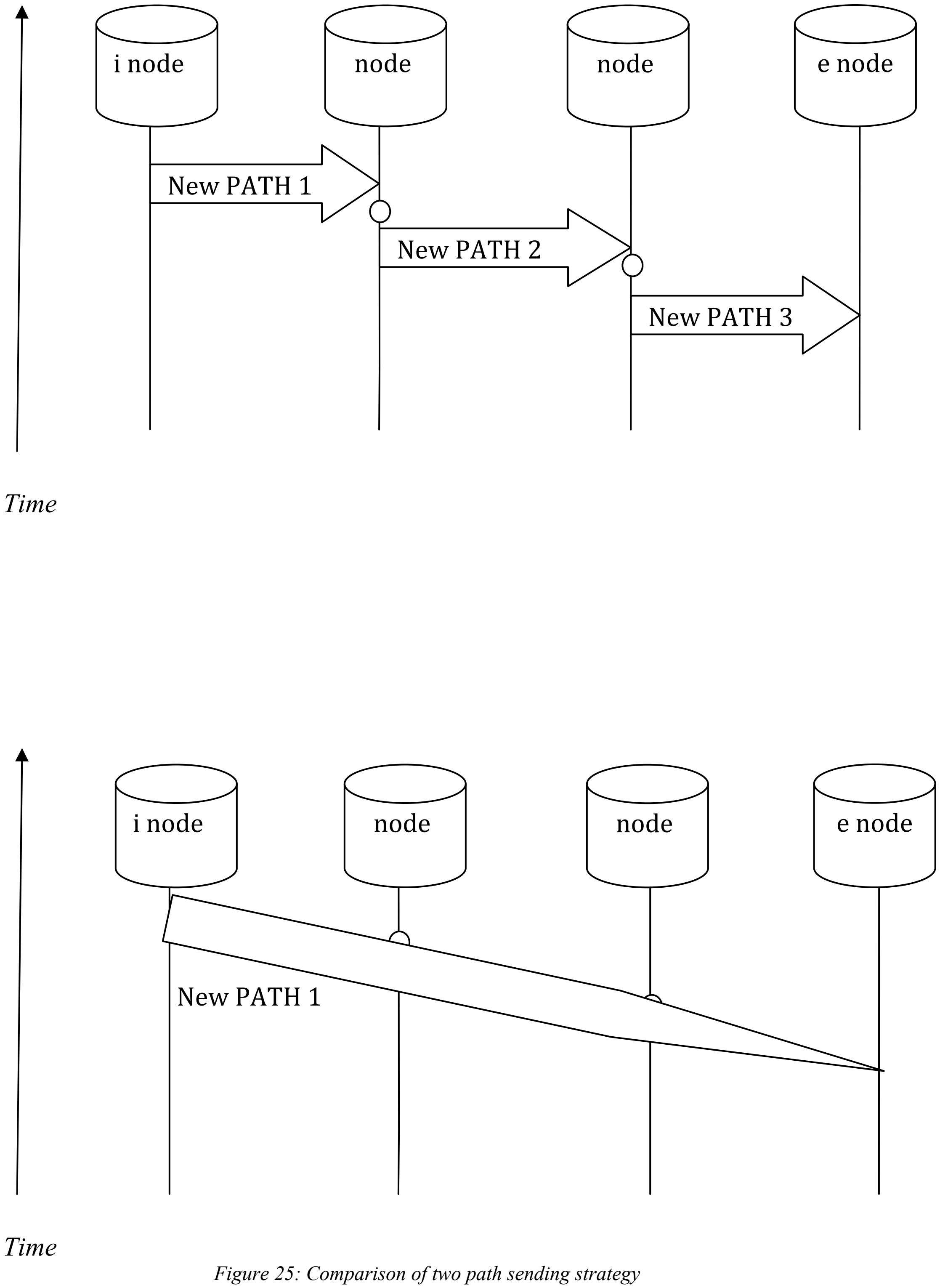


Figure 25: Comparison of two path sending strategy

The critical issue when sending a path message is the time it takes, the time is the sending delay of the path and processing time in the LSR. To fulfill this requirement, I used a hold function when the path instantiates and each time before it began to fill the LIBs of other LSRs on the way towards egress node. In this way the time consumed for this process is equal the sum of all time consumed for each new path in the real system. This implementation reduces the impact of the affect of just one path message.

Network state: Another function in the LSR entity is to update the link state between the two successive nodes in the GST-LSP path if the PATH message performed successfully. The Link state has an overview of the numbers of wavelengths available in the link and which wavelength is allocated for each LSP-request after establishment.

The LSR uses a Boolean variable called “pathestablished” to indicate success or failure of an established link by setting the value to true or false.
When the establishment process is done, the LSP-request object checks the “pathestablished” variable, if it is true, the LSP-request accepts and the filling of the Gasco table starts and LSP continues to be alive. And if the value of “pathestablished” equals false then the LSP- request rejects and terminates.

Fill- in: when the establishment process is finished, the LSR start the fill-in process. It is an entity which carries the LSP id, ingress node, intermediate nodes and egress nod. In the fill-in entity the LSP id compares against the Gasco table to fill the correct table with the correct carried information and allocates that table to be updated later by an update message.
Ingress node: when the fill-in process is performed, a Boolean variable called “control” is set to true. If this variable equals true then the ingress node begins to control the bursts that pass through the GST-LSP. The properties such as burst delay controls by ingress node and saves. The max delay among the number of passed burst registers and the average delay calculated by using the sum of all passed burst’s delays. This registers and sends to the corresponding Gasco-table by instantiation of an UPDATE message.

Creator: after the fill-in process, the creator instantiates by the LSP-request to create a burst. As I mentioned before when an LSP-request is generated by generator, the number of application also determines from the application set.
The creator create different burst for different applications with different properties.
Fist the application type randomly determines from the application sub-set. And based on the application’s type, the properties of a burst determine and the burst sends to the established path (LSP). This process repeats as long as the LSP is alive. The creation is an negative exponential function with a mean equal 0.0004. See table xxxx.
The created burst sends to a queue allocated to the corresponding LSP which is alive.
The ingress node takes the incoming burst from the queue and checks it. It is illustrated as follow:

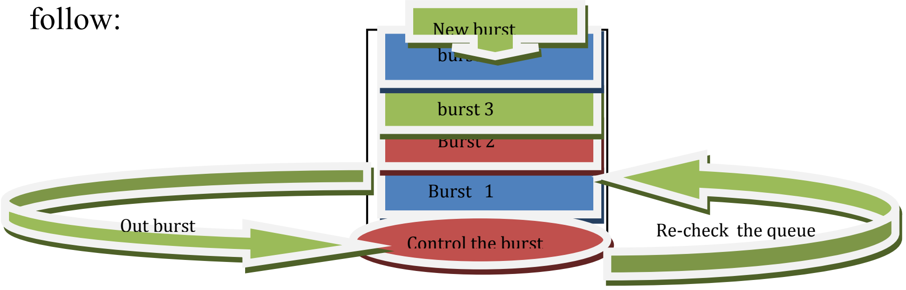


Figure 26: The process of inputting and out- putting of the burst by the burst creator and ingress node respectively

Gasco: The Gasco entity instantiates from the start of the simulation and is running during the simulation time. The Gasco instantiates the gas-tab entity Objects, in this way it contains tables, each belongs to traffic shapes data of a particular active GST-LSP.
In case of SM-LSP, the Gasco finds the best path and locally processes the information and sends the selected path to the ingress SM-LSP path.

Gasco-tab: this entity is actually implemented as a two dimensional array, where the head of the table is filled by the *GST-LSP* properties as *LSPid*, *ingress node*, *egress node*, and *intermediates nodes*. Each row in the table updates by the information such as *peak delay*, *average delay* and *average used bandwidth* in each controlling round. So after a while the *Gasco-tab* for a GST-LSP contains some rows of such information which approximately shows what happens in the GST-LSP.
Each *Gasco-tab* has a unique queue where the update message is putting in it and taking out by the *Gasco-tab*.

UPDATE: the *Update* message is used to send the registered information to the *Gasco-tab*. During the controlling of bursts, when there pass enough bursts and the time of updating is reached the *Update* entity instantiates and carries the registered information. This *Update* message puts in a queue which belongs to the corresponding *Gasco-tab*. Before it puts itself to the queue, it holds a short delay time to realize the sending delay of the message. So while a *GST-LSP* is alive and the bursts of different applications pass through it, the *Updated* message instantiates several times.

8.1 Data Structures:

Network representation: The link between nodes in the network is represented by the maximum number of wavelengths the link consists of. Each time a path message is sent to a node, the value of the wavelength's weight is taken as the wavelength label and the wavelength weight decreases by one to indicate that a wavelength is occupied and can not be used for other GST-LSPs.
Refer to the figure 22 , consider that a path message sent from LSR3 to LSR4 , after reservation of a wavelength the value *2*□ which is in the row 3 and column 4 takes as label, and the this cell value (row 3 and column 4) reduces to *1*□.

LIB : label Information Base exists in each LSR and is represented as a two diminution array as follow:

LSP id	inode	enode	from	to	λ label In	

Table 5: graphical conception of LIB

When a new LSP-request generates, the PATH message sent from the ingress node carries all these parameters and the row number in the LIB of LSR which the PATH is directed to, decides based on the LSP id. For example the information associated to a GST-LSP with id equal 4 fills in the row number 4.

Labels:

In the GST-LSP establishment of the label is a lambda label which refers the wavelength reserved for this LSP. By following the allocating strategy explained in the Network representation paragraph, a GST-LSP can have a different lambda label depending on the available wavelength weight in the network array between each two successive nodes (a link). So the allocating of a lambda label is restricted and limited, since the number of available wavelengths in each link is limited in the simulation mode.

Gsaco-tab: The data in a Gasco-tab entity object arranges and schedules as in the following:

Head →		LSPid	ingressnode	intermediate nodes	egressnode	
Body →		Peak delay	avr. Delay	avr. Bandwidth	gaps nr	
	Upd1 Upd2 . Upd n					
	$\sum a_i\delta(i)$					

Table 6: Graphical conception of the Gasco-tab structure in the simulation

The head of each table checks to find the paths which can be used for SM-LSP. When there are several numbers of such paths, the next step is to compare Peak delay and other parameters stored in tables which belong to the GST-LSP paths which can potentiality to be used for SM-LSP.

Application: the applications are represented as numbers in the simulation. For example the VoIP is represents by 1, Video confronting by 2 and etc.

Burst size: the size of each burst randomly draws from an interval of minimum and maximum numbers. Different applications have burst with different size intervals.

Burst duration: deviation of burst size by requested bandwidth return the burst delay.

Number of gaps: The numbers of gaps are equal to the number of bursts that are created in the controlling time up to the updating time. Since I assumed that there is always some delay between creations of bursts.

8.2 Extension of simulation: After the simulation of the Gasco and its updating, the next process is to extend the simulation in order to simulate the SM-LSP request and its acceptation by checking the Gasco table to find a proper path. For this propose I add a column in the head of Gasco_tab to be filled with a Boolean variable. If the Boolean is false (or 0) it indicates that the GST_LSP which this particular Gasco_tab belongs to, is not used by any SM traffic. If the gaps in a GST_LSP are used for SM traffic, the Boolean variable returns to true (or 1) to indicate that the SM traffic utilizes the gaps in this path.

SM-LSP_request:

It is an entity containing the ingress and egress as in-parameters. This entity generates every 2 seconds. The ingress and egress draws randomly. The SM-LSP_request instantiates a message entity which names info_req. this entity puts in a queue which checks continually. This message carries the ingress and egress which belong to the SM_LSP_request.

Gasco functionality: The Gasco entity is extended by a processing function to take out the queued info_req message and process it. The processing includes comparing of the ingress and egress nodes against nodes in every Gasco_tab in the Gasco to check if these are in there. If these nodes are in a Gasco_tab, the GST_LSP which the Gasco_tab belongs to, is assumed as a candidate to the SM-LSP. The Gasco checks through all Gasco-tabs to find all GST_LSPs which have potential to be used also for SM-LSP. If there are several candidates of such Gasco_tabs, the one with the smallest mean peak delay is chosen for the flowing of the SM packets for the requested SM_LSP.
The Boolean variable in the head of the chose Gasco_tab returns to true to indicate that the gaps in this GST_LSP is used for the SM packets and is therefore occupied.
So during processing of a new info_request message, the occupied GST_LSP or in a better word the Gasco_tab with Boolean variable equal to true is not on the cards and will not be checked.
I used a hold function in the Gasco during the finding of the best Gasco_tab, in order to mirror the processing time in the real Gasco systems.

Generating time of SM_LSP_request:

I implement a generator to draws random ingress and egress nodes and generate a new SM_LSP_request every 2 sec. The generating begins almost in the middle of simulation time. This is because to ensure and guaranty that there are a sufficient number of established GST_LSPs for the Gasco_tab in the Gasco. The start time of generator ensures that the system is in its consistent state.
Several times and functions were tried to find out the interval between arrivals of SM_LSP requests. A Poisson process was finally chose with a mean time equal to 1,5 sec.
This gives almost a total nr of 15 SM_LSP request in a time of 30 seconds. I mentioned that the simulation time of the system is 1 min or 60 seconds

GST_LSP request vs SM_LSP request:

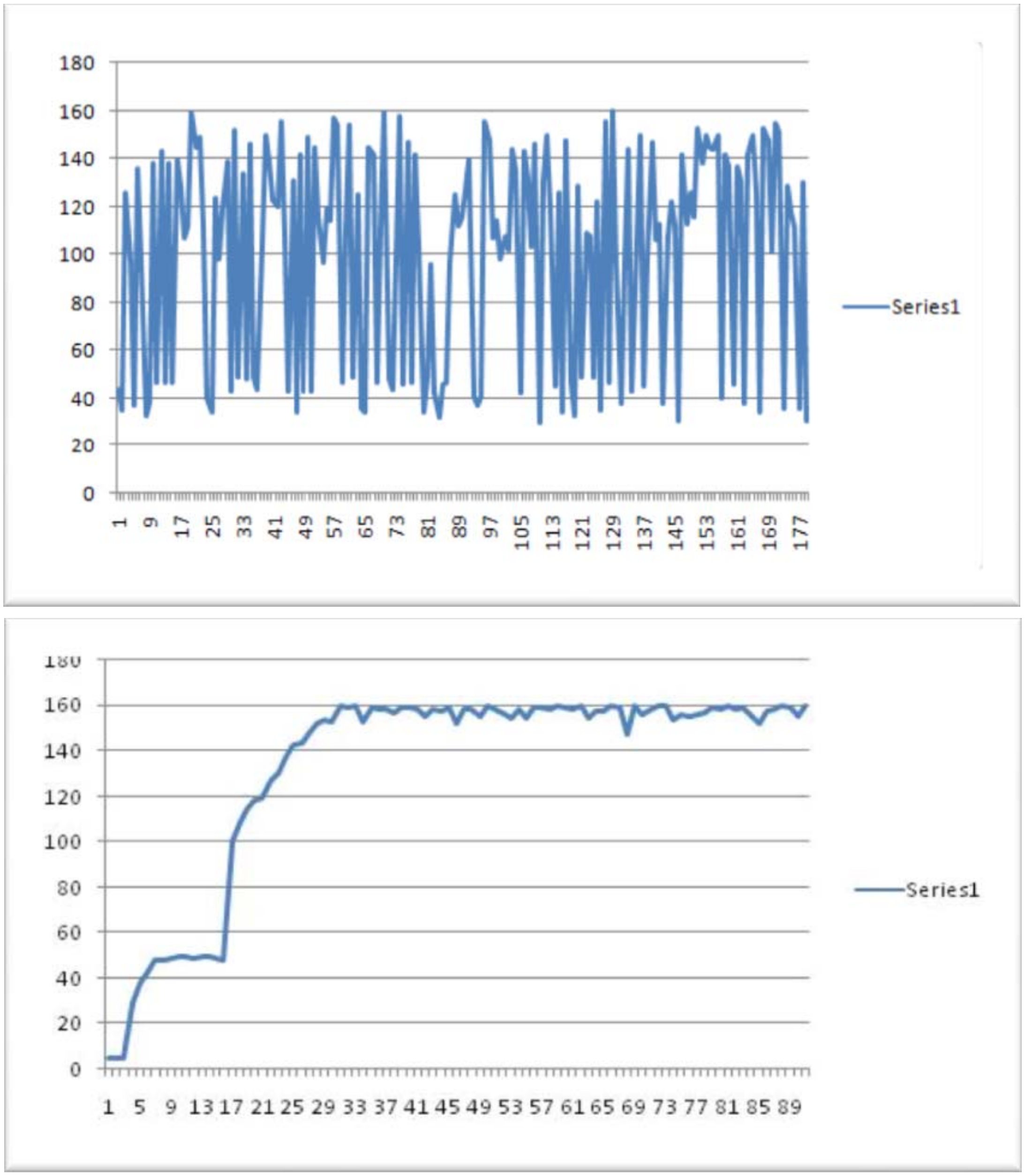
Since in this simulation the process of GST_LSP establishment and Gasco updating was important and not the number of GST_LSP requests, I assumed that there will be a maximum number of 12 succeeded GST_LSPs in the simulation. And this amount of GST_LSP requests are generated in the first half of the simulation with a time interval between each generation. This is to ensure that the network is used by some GST_LSPs in the starting time of the SM_LSP requests.
The generating of SM_LSP requests begins in the second half and continues to the end of simulation.
In the next section, the simulation results will be studied to find out the peak delay variation in a GST_LSP and its impact and influence to the finding process of a SM_LSP.
The importance will be the study of the acceptance and rejection rate of SM_LSP requests by using Gasco.

9.0 Results of Simulation:

The main objective of this study was to understand the Gasco updating in order to have an overview of the GST-LSP path states. The data which reflecting path utilization is controlled and registered locally and an updating action is triggered to carry these registered data to the Gasco.

The simulation result of some GST-LSPs establishment and their resource utilization during the life time is recorded and is shown in the following:

Figure 27 shows the peak delay stored in Gasco table for LSP1 from the start of simulation nr 1 and until the end of this simulation. The curve in the first figure illustrates that the peak delay varies approximately between 40 and 160 micro second (Y-axis).



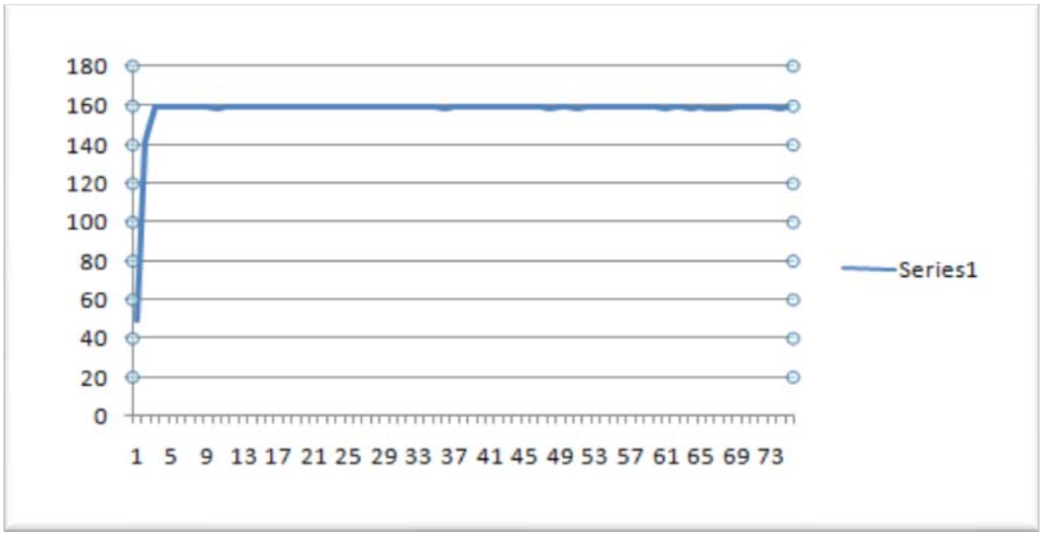
X-axis= updating number

Y-axis= delay in micro second

Figure 27: Peak delay recorded by different updating rate in first and second graph

This coarse variation of peak delay is because of hyper updating rate. The time between updating is very low and therefore the recorded peak delay belongs to the burst's duration of different application in each updating time. This will not give a good estimate of the peak delay. In order to improve the estimation the updating rate should decrease. This decrement gives the possibility to check more bursts between two successive update.

LSP1



X-axis= updating number
Y-axis= delay in micro second

Figure 28: Some samples of Peak delay

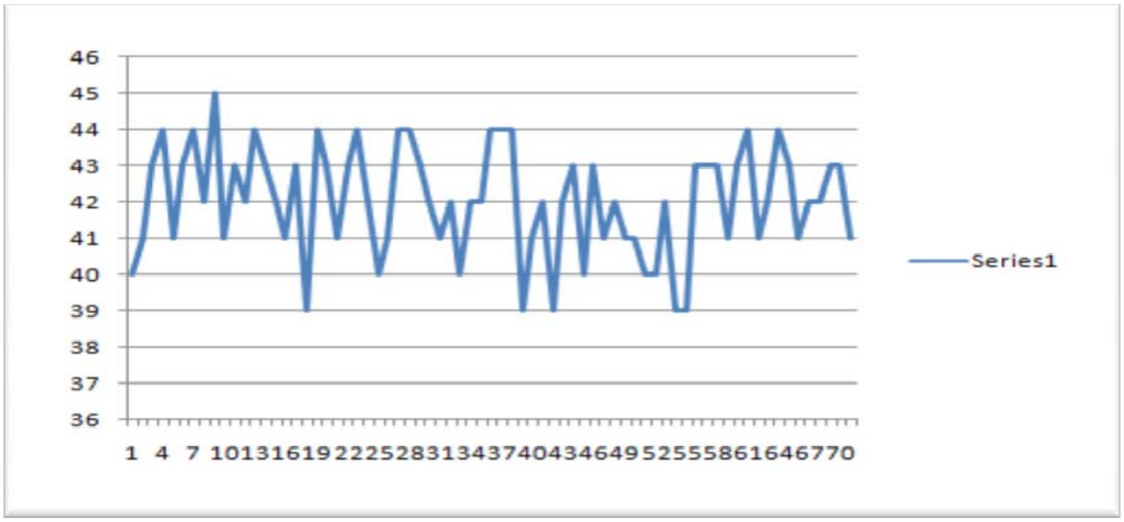
Peak delay properties of LSP1:
Mean: 158.7652
Standard deviation: 10.46408
Confidence Interval: 0,061188
158.8264055>*Mean*>158.7040293

The decreasing of the period of time between updating to a scale of almost 0.5 second gives more stabile and exact peak delay. This is obviously from the above figure and statistics calculated for LSP1. The mean value and standard deviation is given, with a 95% confidence interval.

This high delay time with very low variation reflect that during the life time of LSP; there was either a smooth flow of all kind of application bursts or just burst of one application that has this high burst delay. A trade of between peak delay curve and average delay curve (figure 29) make us sure about the variety of applications in the LSP.
Figure 29 shows that the average delay varies between 40 and 110 which reflect that there was a smooth utilization of LSP by different application during life time of LSP.
If the time between updating decreases, the peak delay graph gets a shape which is near to the average delay graph. It means that the controlling time is shorter and hence the probability that bursts of all kinds of application flows and checks in every controlling period is less, which will results in a wide peak delay variation in the graph. If the time between each updating increases, the probability that the selected peak delay belongs to the application with highest delay amount the application set is very high and near to
100 %. This results in a very straight and smooth line in the graph plot. See figure 28. Since the peak delay shows the maximum time a SM packet are going to wait before it sends to the path, it is important that the Gasco have a good picture of the peak delay and therefore a reasonable controlling time between each updating is chose to be about 0,5 to 0,85 second.

The peak delay does not indicate that all SM-packet should wait a time equal peak delay stored in DB, but this is a maximum waiting time a packet could stay before it sends to the path. The average delay illustrated in figure 29 shows that it is possible that a packet waits at least 40µs.

LSP1



X-axis= updating number

Y-axis= delay in micro second

Figure 29: Average delay

Average delay properties of LSP1:

Mean:	*41.66087*
Standard deviation:	3,783359
Confidence interval:	0,022123

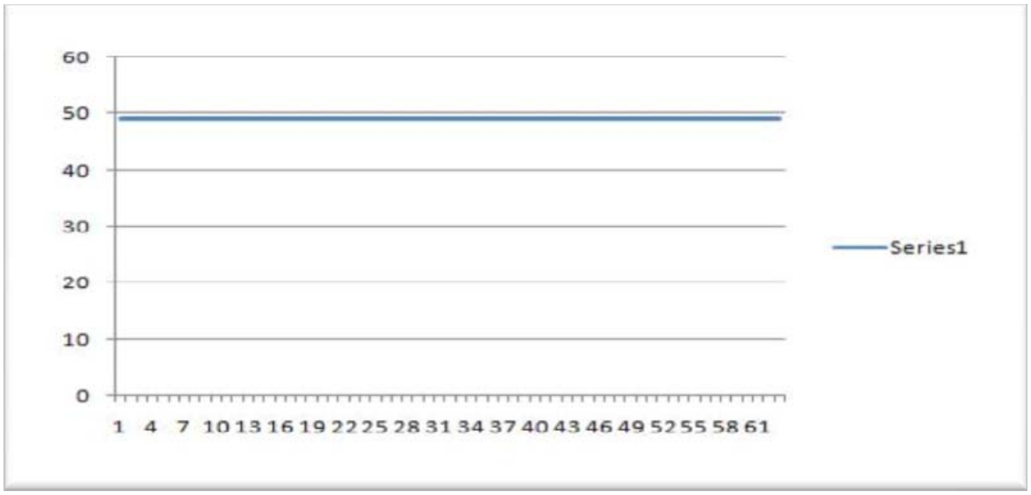
41.68299> *Mean* *>41.63875*

As the figure 29 illustrate, the traffic flow in some period of GST_LSP was high and crowded and flowed bursts had high delay. See upper bounds in the graph. But most of the bursts had delay between 40 and 45 µs.

During checking of tables in Gasco for choosing the best path, if there are several tables with same main peak delay, the comparison of average delays in the candidate tables contributes to a reasonable selection.

Another sample of data collected and stored in a Gasco table is represented in the next page. This belongs to a GST_LSP in another simulation.

LSP4



X-axis= updating number
Y-axis= delay in micro second

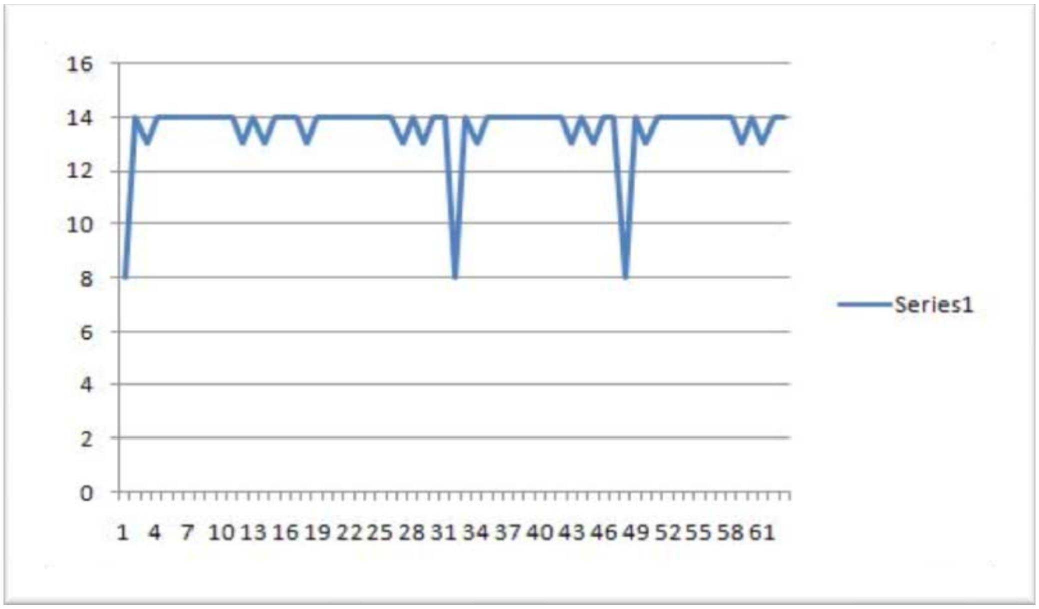
Figure 30: Peak delay

Based on the figure we can understand that the bursts in this GST_LSP were not of all kind among the application set, but some of the application. This is because the recorded peak delay in all updating rounds is less than the duration of burst with maximum length amount the application set. The recorded peak delay is exactly the same in all updating rounds and is equal 49. It looks suspicious that the path contains just the video streaming burst. See table traffic model. But the average peak graph shows that the path has belonged to not just the over mentioned application, because the average delay fluctuates.

In a case of SM_LSP request, this GST_LSP will be choice than the LSP1 represented over.

Since the peak delay is identical in all rounds, so the statistics properties are not mentioned here.

LSP4



X-axis= updating number
Y-axis= delay in micro second

Figure 31: Average delay

Average delay properties of LSP1:

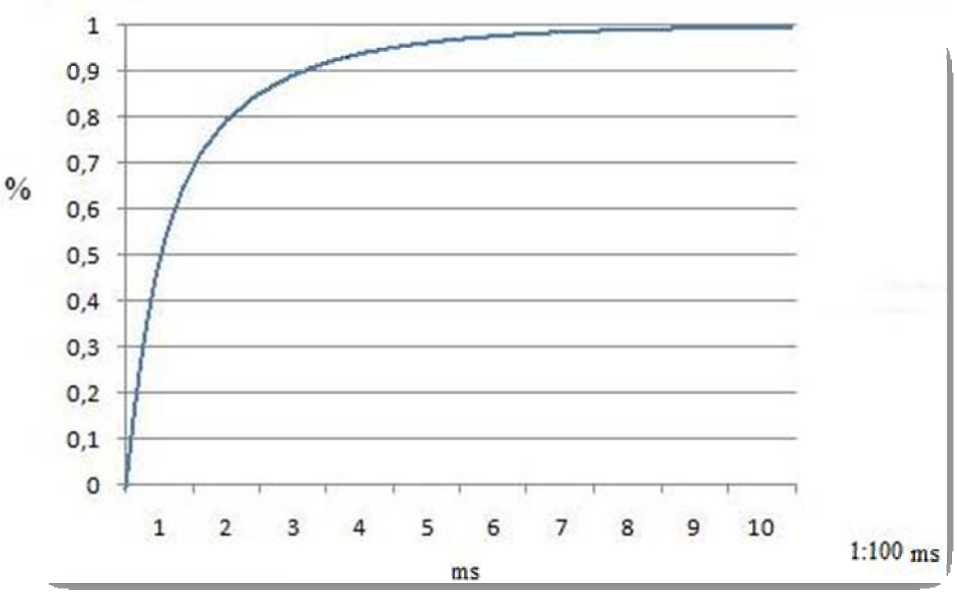
Mean:	13.3125
Standard deviation:	1.305783
Confidence interval:	0.0227098

13.33521> *Mean* > 13.28979

The average delay graph is more stabile in compare to the corresponding graph of the LSP1. This is because the number of applications is less than the amount of application in the LSP1. It's obviously also from the standard deviation and confidence interval.
There will pass more SM-packets through the path, when a SM-LSP path is established by using the gaps in this GST_LSP. It's because the waiting time is short and therefore faster the burst out puts from the buffer.

9.1 Updating rate and link state:

Updating rate or the time between successive updating is important in order to have a correct view of the link or path state. The parameters and kind of information which is carried to the Gasco by each updating also dose matter to the rate of updating. For example in this report I suggest that the most significant parameter is the peak delay which is proportional to the maximum or longest duration of bursts in the GST_LSP path. I found that to have a correct view of the peak delay, the updating rate should decrease and time between updating increase such that the ingress node checks through more numbers of burst. This is to find the longest burst with respect to its duration amount all bursts passed through node in the controlling time. The following graph shows the relationship between updating rate and the correctness of information in percent (peak delay discovery).



X-axis= time in milli second
Y-axis= correcness in %

Figure 32) updating rate and correctness of the peak delay

This figure is obtained by a test where I changed the updating rate and checked how the Gasco is updating by information as peak delay. I run several simulations by different updating rate. The result of test showed that a good updating rate is from 0.4 second (400 ms) and up to 1 second. It is because the Gasco get a correct view of peak delay with a correctness of almost 85 % by using 0.4 as updating rate. But I used 0.5 second (500ms) as updating rate since the correctness is over 90% by this rate and the Gasco table use less storing place in the Gasco memory.

9.2 SM-LSP request: The generating of SM-LSP request is performed in 10 simulations and the acceptance and rejection rate is as followed:

```
**********************************************************
                           T O T A L
                       S I M U L A T I O N
                          R E P O R T
                       Based on results of 10
                         Sub-simulations
**********************************************************

sum_SM_req=140
sum_SM_rej=34
accept_percentage ≈75.71%
```

As I mentioned previously the SM_LSP request begin to generate at the middle of simulation time of each sub-simulation. These generates by a Poisson process every 2 seconds and a number of request become almost 12 to 15[1]. The acceptance of request is depending on to the number of the running GST_LSPs, the ingress and egress nodes of SM-requests. It is also valuable to remain that the accepted SM paths also do not terminate during the simulation time.

[1] Total sub-simulation time=60 second
Middle sub-simulation: 60-30=30
Generating rate :2 s → 30/2 = 15

10.0 Conclusion:

With use of Gasco and SM_LSP packet labeling in the control plan as a centric solution to find and establish the most proper path for the SM packets in the OpMiGua network, some aspects of traffic engineering can be obtained. These aspects are reduction of network topology where just the links and nodes which are in the GST_LSPs are checking to find a SM_LSP. This also brings the constraint based routing. The constraint is that there should be at least a GST_LSP in the network. And not all nodes and links in the network are playing roll. The Gasco is updating by parameters as peak delay, average delay and an approximation of number of gaps in each GST_LSP.

In this report I concentrated on the peak delay and average delay. Concerning about gaps, I assumed that there is a gap between every burst in the path and therefore I didn't concentrated on gaps .The peak delay is the longest time a SM packet should wait before it sends to the SM_LSP path.

With the argumentation above, in addition to the routing and establishment of GST_LSPs the updating process of Gasco was simulated. The result showed that the updating frequency do not need to be fast and an updating speed of 0.5 second is sufficient to give a good and feasible view of the traffic shape with respect to GST burst duration in a GST_LSP. The maximum burst duration in a controlling period between two updating process registers as peak delay. This updating frequency is obtained based on simulation is an advantage with respect to number of updating and also storing space in the Gasco memory.

Another aspect of simulation was to find out the successfulness level of the SM-LSP request. The simulation gave an acceptable result of SM_LSP request acceptance. The acceptance is proportional to the number of GST_LSPs in the network and SM-LSP requesting frequency. A frequency of 2 second was feasible. It was obtained by trying different time between SM_LSP requests.

As it was suggested in the simulation, the Gasco should be able to process the SM_LSP requests locally and fast enough to return the best path or link for the requested SM_LSP, instead of sending a copy of all GST_LSPs data which is stored in Gasco.

The structure of traffic in the simulation was based on application and their proper bursts.
A drawback of this simulation was that the traffic of all applications generate similarly with the same distribution function, but in the real system different traffic have different flowing rate and speed.
Each application had a particular burst with respect to the duration interval. The reason that the updating is not so frequently is this property of the simulation, namely application based traffic structure. This was an advantage which makes the outline of the simulation easier and gave a good result for updating speed.

10.1 Further Work

The simulation can be extended by including some other properties of the OpMiGua. A possibility is to develop and add the gaps properties which appear in the GST_LSPs. Gaps duration, number of gaps with respect to the traffic feathers and distributions. The extension can include the possibility to tear down a path of both kind and increase the simulation time in order to adapt it more with the real system's feathers.

The application can classifies based on the distribution properties and see how the peak delay changes during the connection time of a GST_LSP.

References

[1] Vlachos, Kyriako's "Photonic Network Communications, Volume 13, Number 3, June 2007"

[2] E.Mannie, "Generalized Multi-Protocol Label Switch (GMPLS) Architecture",IETF RFC 3945, Oct.2004

[3] Daniela Vieira Cunha, Graca Bressan, Generalized MPLS – An Overview, Sao Paulo-Brazil

[4]" Control Plan for OpMiGua" NTNU, 2007 M. Yaser Rahmati

[5]Transparent Optical Switches: Technology Issues and Challenges
G. Ellinas*, J. Labourdette*, J. Walker, S. Chaudhuri*, L. Lin, E. Goldstein, K. Bala*
*Tellium Inc., 2 Crescent Place, Oceanport NJ 07757

[6]Hybrid Optical Network Architectures: Bringing Packets and Circuits Together, Christoph M. Gauger,Erik Van Breusegem,Paul J. Kühn,Mario Pickavet ,Piet Demeester

[7]http://www.diva-portal.org/ntnu/abstract.xsql?dbid=1032

[8] Telektronikk 2.2005
http://www.telenor.com/telektronikk/volumes/index.php?page=overview&id1=66&select=05-09

[9]Test-Bed Demonstration and Performance
Martin Nord, Steinar Bjørnstad, Oddgeir Austad, Vegard L. Tuft, Dag Roar Hjelme, Aasmund S. Sudbø, and Lars Erik Eriksen

[10] MPLS, Technology and Applications , Bruce Davie Yakov Rekhter, 2000 by academic press

[11]
http://www.webopedia.com/TERM/M/MPLS.html

[12] http://www1.alcatel-lucent.com/doctypes/articlepaperlibrary/pdf/bnbstraffic.pdf

[13]
http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/120newft/120limit/120s/120s5/mpls_te.htm#42292

[14] "Generalized MPLS - An Overview" **Daniela Vieira Cunha, Graca Bressan** Department of Electrical Engineering - POL1 Escola Politkcnica, University of SZo Paulo **(USP)** S5o Paulo - Brazil

[15]Generalized Multiprotocol Label Switching: An Overview of Signaling Enhancements and Recovery Techniques, *Ayan Banerjee, John Drake, Jonathan Lang, and Brad Turner, Calient Networks*
Daniel Awduche and Lou Berger, Movaz Networks Kireeti Kompella and Yakov Rekhter, Juniper Networks

[16] Lightpath Routing for Intelliaent Optical Networks U I Shenzhen Zhang and James Fu, Sorrento Networks Dan Guo, Turin Networks, Inc. Leah Zhang, Photuris, Inc.

[17] http://www.juniper.net/techpubs/software/nog/nog-mpls-logs/html/status-messages18.html#1037087

[18] http://snad.ncsl.nist.gov/glass/doc/pdf/gmpls/RSVP_Message-English.pdf

[19] http://www.cisco.com/en/US/docs/voice_ip_comm/cucm/srnd/5x/50nstrct.html

[20] http://www.essexcorp.com/News/terabit.htm

[21] "Optical Components for WDM Lightwave Networks", MICHAEL S. BORELLA, MEMBER, IEEE, JASON P. JUE, DHRITIMAN BANERJEE, BYRAV RAMAMURTHY, STUDENT MEMBER, IEEE, AND BISWANATH MUKHERJEE, MEMBER, IEEE

[22] Implementation of a GMPLS-based Network with End Host Initiated Signaling,
Xiangfei Zhu, Xuan Zheng, Malathi Veeraraghavan , University of Virginia
Zhaoming Li, Qiang Song, Ibrahim Habib Nageswara S. V. Rao
City University of New York Oak Ridge National Laboratory

[23] http://www.devicescape.com/docs/smap/AdminGuide/QoS.php

[24] TTM4110 Pålitelighet og Ytelse med Simulation , peder J.Emstad, Poal E. Heegaard ..

Appendix A

A section of previously project

4.1 Structure of a control plan for OpMiGua

After studying of OpMiGua and (G) MPLS it was obvious that GMPLS has the ability to contribute the control of OpMiGua. Fiber switching capability, lambda switching capability and packet switching capability are the feature of GMPLS which can be usable in developing a control plane for this special kind of network.

The OpMiGua network has special feathers and goals. So to construct a feasible and suitable control plane for it first we should consider these features. One of the most important specifications of it is that it transmits two kind of traffic namely GST and SM.

Therefore building a control plan must be based on this specification.

4.1.1Control plane parts:

4.1.1.1 LSP: GST and SM traffic has different requirements and hence two different kind of LSP is needed to forward these traffics in an efficient way in this network.GST traffic is the one with highest priority and good Qos requirements, therefore all the resources like bandwidth, wavelength and components like switches, fibers first allocated to them when both GST and SM traffic need them. So we can compute and set up a fixed and reliable LSP path by using of available and necessary resources through network. For this purpose we can use the capabilities existing in GMPLS. By using GMPLS we can set up a LSP in fiber level, waveband level or wavelength level. The optimal level of switching path actually depends on the amount of a specific GST traffic. For example if a GST flow from a certain origin node to a certain destination node is so high that it needs the bandwidth capacity of fiber so the LSP can be setup at this level. But a 10 billion digital bits can be transmitted per second along an optical fiber link in a commercial network which is a huge capacity related to a GST traffic flow.

If the fiber level is chosen for path setup so the GST packets can propagate in what wavelength among the total wavelengths in fiber and therefore many unused capacity are blocked. In such a case there will be an inefficient use and waste of resources which is in contradict with one of the OpMiGua goals. The optimal chose among the rest possibility is wavelength level of switching. So by obtaining a wavelength as a label for GST traffic along a path since the traffic amount is not bigger than bandwidth of a distinct wavelength, other available wavelengths in a fiber can be used for GST traffic with different destination or applications type. In this way the bandwidth resource is utilized in a feasible and optimal manner.

To compute the shortest LSP path it does need information like available capacity, network topology and administration constrained. This information is provided and flooded by some routing protocols like OSPF or IS-IS. Both of them are useable and enhanced in GMPLS. In the signaling phase the computed LSP is setting up by some signaling protocol or more specific the GMPLS signaling protocol namely RSVP/TE protocol. For more explanation see **RSVP/TE** and **OSPF** subsections in previous section.

The LSP setup is illustrating in a more conceptual way in the figure 5(later).

4.1.2 Setting a LSP for SM traffic

There is a lot of alternative to take when developing part of the control plane which controls the SM traffic. Developing this part of control plan is more complicated but we can start with the same option taken for GST.

With this option we consider the wavelength (lambda) as a LSP path level, since the amount of a SM flow is not more than GTS flow. By assigning a distinct and exclusion wavelength to a SM traffic through OpMiGua network we can establish a LSP path as the same way as for GST packets. But

setting up a LSP in this way has some disadvantages and drawbacks. At the first in this method both GST and SM packets have the same access right to the resources and therefore no difference between GST and SM classes. Prioritization of GST traffic becomes meaningless. To overcome this, the LSP can establish through an already used wavelength switching GST traffic. In this way SM packets just use the gaps which appear between two successive GST packets and if there is no enough gap in compare to number of SM packets, the packets can be send to buffer and stay there until a gap is available. Switching of SM traffic in such a way dose not require any wavelength converting which can be an advantage but at the other hand it needs larger buffer. Sending and storing of SM packets in buffer is waste of both time and resource. The gaps in other wavelengths are underutillaised which is another drawback with this strategy. One another obviosed disadvantage with this method is that the statistic multiplexing capability of OpMiGua is not fully used.
How about the gaps? To label a flow based on just gap is also a suggestion, but the GMPLS not equipped with this kind of label switching. So to use this as label it requires extension of GMPLS which is out of scoop of this project. If the GST packets had fixed length and fixed transition rate we could consider the gaps as timeslots and in such situation timeslot labeling could be a suggestion. Effective utilization of the gaps is a serious affair in the OpMiGua, so what is the best switching method of SM traffic with respect to underutilized gaps? To take the gaps in account and fulfill the static multiplexing I propose a structure for control plane which is a little different from other control planes for optical networks. Because the network consist of nodes designed for two type of traffic's switching namely GST and SM traffic, it looks that the control plan should also consist of two part . One part for circuit switching which was discussed in above (LSP for GST) and another part for SM traffic which is described later in the paper. A hybrid control plan for a hybrid optical network!
The details of the proposed structure are as following.

4.2 Gaps Statistic Container (Gasco)

In OpMiGua node a SM packet can transmits through any wavelength which has gap, and this is why it exist wavelength converter in the OPS part of the node. Any wavelengths in a fiber can be used for switching a specific SM packet. When the destination node is computing the LSP for a SM traffic flow which is directed to it, it should have an overview of the wavelengths with most gaps in a fiber. A statistic of available gaps is needed when computing the best and shortest path through the network. This overview can be obtained in two ways. One is that each node in the network broadcast or floods to other nodes its Owen statistic of gaps which appear on the wavelength through it in a period of time. As mentioned in previous chapter the node is equip with an electronic detector to detect the gaps. The other consideration is to have an entity which has contact to all nodes, and the nodes send the statistic to it periodically. So by using an entity which contains a statistic of Gaps over the whole network, the organizing will be more centralized in contrast to flooding of statistic to all nodes. This entity that I call it for Gasco can play a role like a virtual database or perhaps be a physical one and have to kind of messages. One for updating the database namely UPDATE message and one which called for REQUEST for obtaining information from it by a node which want to compute LSP.

4.3 Labeling of SM traffic

So in the next step when the destination node get information includes gap's information via flooding or Gasco it can compute the optimal LSP. This LSP is established through a link with most density of gaps. If we take the fiber as a label seeing that the SM packets are distribute over all wavelengths placed on the LSP fiber. Using a fiber label limits us to switch just one kind of SM traffic flow at a time. The other SM traffic flow with eventually same destination should wait until the path is free or they should switch through another LSP. To overcome this we can send both SM traffic flows at the same LSP simultaneously by using a better granularity of switching namely a logical packet label hierarchy switching.

Before explaining how we can optimally use the available resources with better granularity of switching, the SM traffic class should divide in subclasses based on their importance and priority.

The prioritization can be determined based on for example application type of SM or Qos requirements or other feature which is worth to be taken in consideration under the prioritization. For simplicity of the case we can start with just two kinds of subclasses as followed:

- SM with high priority denoted by SM_H
- SM with low priority denoted by SM_L

I remind that the SM traffic is distinguish from GST traffic based on polarization detected by a splitter early in the physical layer to avoid packet type detection later on the above layers. So how we can detect and separate the SM subclasses. For this propose we can use Qos field in the packet's header to mark the SM type.

During switching of SM traffic the node sends SM_L packets through wavelengths with the least number of gaps and limits this type of traffic to these wavelengths. And let the SM_H use the wavelength with most gaps and all other free gaps in the link. This is a better granularity of switching .This strategy is similar to the one taken in the storing process in the hard disk in computer where the smallest data are stored first in the shortest free spaces and the large space are left to be used for large data. But the difference is that in this case the traffic flow is not pre-determined and fixed in contrast to data storing in hard disk. So classification of SM traffics makes it possible for us to use the packet label switching. The SM packets are labeled based on their priority and forward through the LSP on the wavelength with high density of gaps or lowest density of gaps depending on label type or more exact subclass type.

Here we could use of waveband label such that the wavelength with less GST packet or in another word with highest gap density were bunched as one waveband and the rest waveband in the fiber was bunched as another waveband, but this require successive wavelengths in a bunch. This requirement is not fulfill in all cases therefore using of packet labeling is a better candidate. What we should have in mind is that labeling happened in a packet label hierarchy manner in a logical way such that we just setting up a LSP for both SM subclasses. The reason for this policy will be discussed later in the Local Policy subsection (Ratio change).

4.3.1 Local Policy

Each node along the LSP should be able to get information from Gasco by sending a request massage to it when it is necessary. After establishment of a LSP for whole SM traffics flow directed to a specific destination, the node should be able to separate between SM subclasses such that it send the SM_L to the wavelength with lowest gaps and the SM_H to the wavelength rest of wavelengths. One case where each node should be able to react in a proper way is change in streaming rate of traffic. It is described in below.

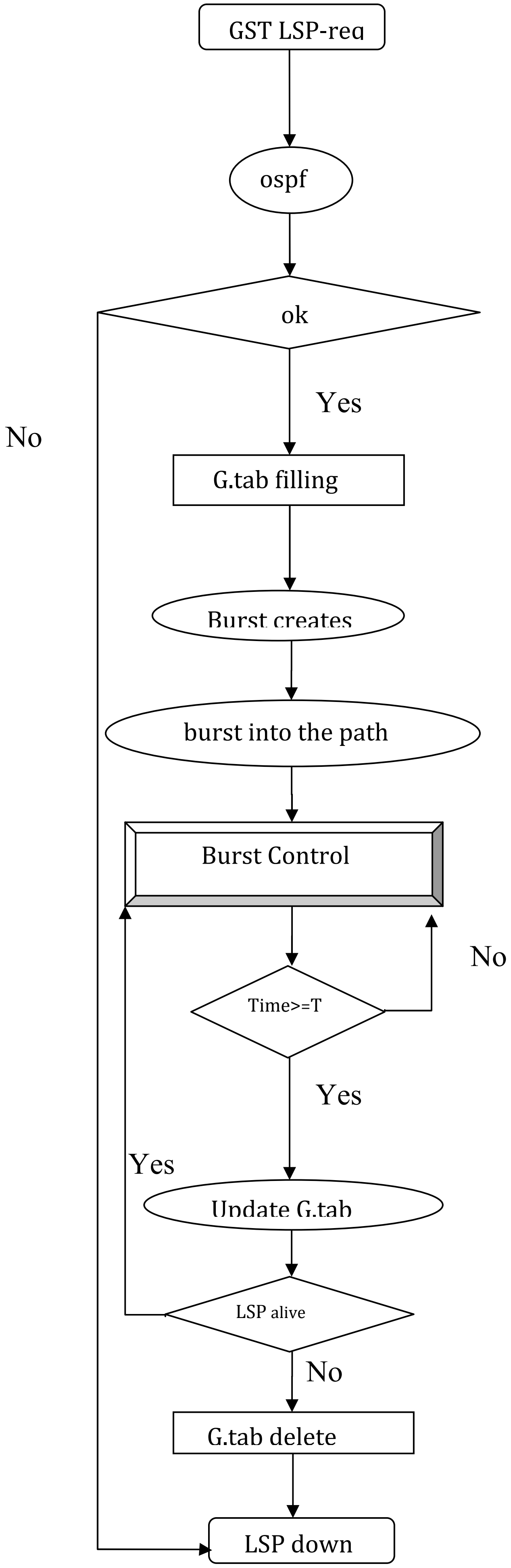
4.3.1.1 Ratio change:

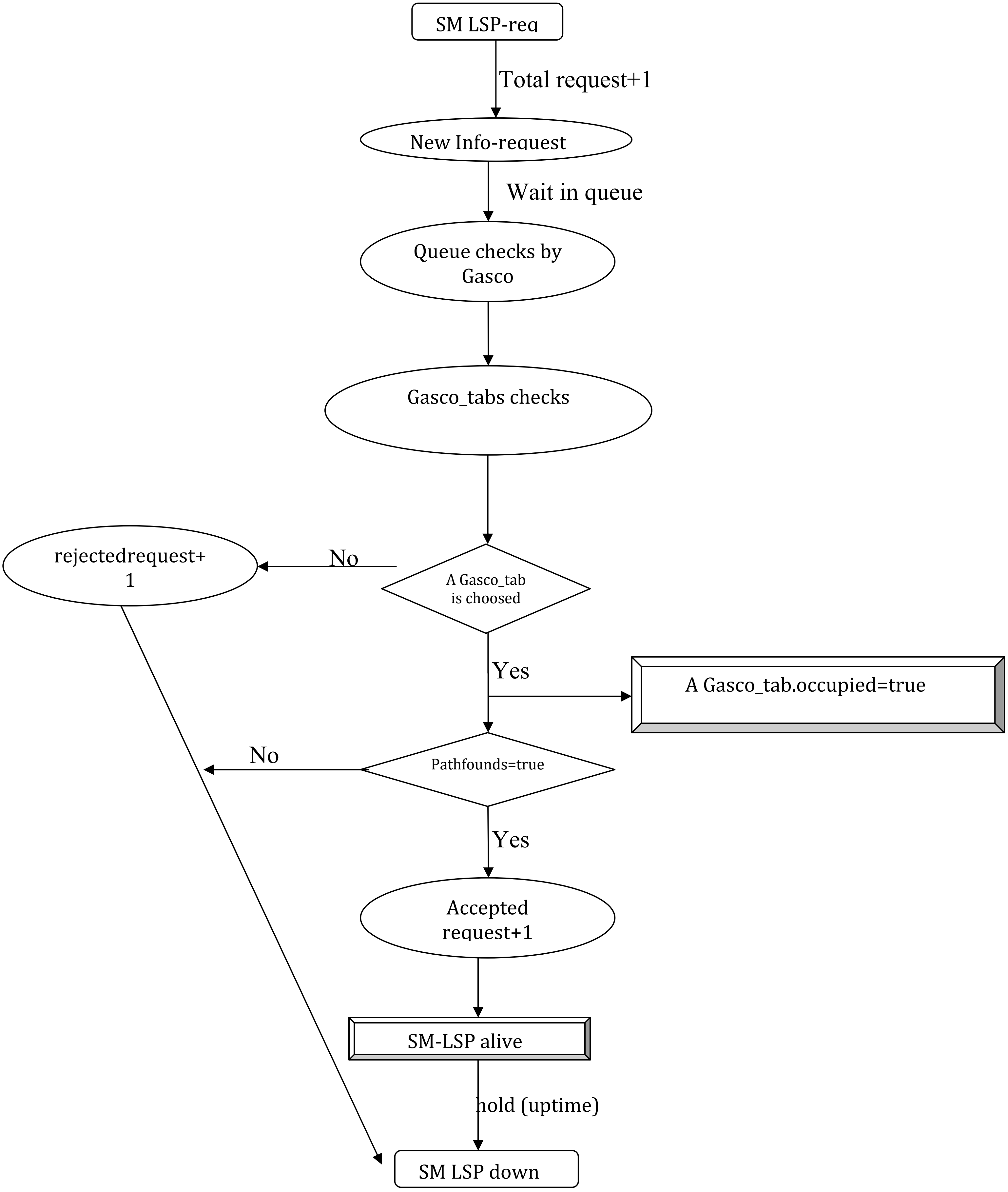
It is possible that streaming rate of the traffic is increasing in the LSP during switching of traffic such that the amount of gaps is not sufficient for flowing of SM_H traffic. In such case the node force the SM_L traffic to go to buffer and the SM_H traffic permits to use these free gaps which was occupied by SM_L packets. This policy lets the traffic with high priority to utilize resources. The wavelengths with fewer gaps which are used for SM_L packet switching in normal situation act as a backup path for SM_H packets. This is why I suggest using two lapels for each packet type at the same time.

If the SM subclasses is more than two, the one with less priority is forced to be buffered and the next subclass take over the free wavelengths and give it's original wavelength to the traffic with highest priority namely SM_H**.** It is a kind of step by step relinquishment of resource. It appears three cases of forwarding. In the best case all subclasses are allocated a part of available resources based on their priority and flow simultaneously through the path. In the worst case because of limited recourse availability just the subclass with highest priority is allowed to use all the available resources and the other should be buffered and waiting or be lost.

The last case where one type of traffic is stored and the rest are flowing can be called good case or better case.

Appendix B


GST LSP-req
ospf
ok
Yes
No
G.tab filling
Burst creates
burst into the path
Burst Control
No
Time>=T
Yes
Yes
Undate G.tab
LSP alive
No
G.tab delete
LSP down


SM LSP-req
Total request+1
New Info-request
Wait in queue
Queue checks by Gasco
Gasco_tabs checks
rejectedrequest+ 1
No
A Gasco_tab is choosed
Yes
A Gasco_tab.occupied=true
No
Pathfounds=true
Yes
Accepted request+1
SM-LSP alive
hold (uptime)
SM LSP down

Abbreviations

OpMiGua	Optical Migration Capable Networks with Service Guarantees
MPLS	Multi Protocol Label Switching
GMPLS	Genera
GST_LSP	Guaranteed Service Traffic_ Label Switching Path
SM_LSP	Statistically Multiplexed Service_ Label Switching Path
SM	Statistically Multiplexed Service
Gasco	Gap Statistic container
SONET	Synchronous Optical Network
OPS	Optical Packet Switching
ORION	Overspill Routing in Optical Network
LED	Light Emission Diode
TDM	Time Division Multiplexing
WDM	Wavelength Division Multiplexing
OEO	Optical to Electrical to Optical
MUX	Multiplexor
DEMUX	De Multiplexor
DWDM	Dense Wavelength Division Multiplexing
IP	Internet Protocol
SDH	Synchronous Digital Hierarchy
QoS	Quality of Service
ATM	Asynchronous Transfer Mode
OXC	Optical Cross Connector
CS	Circuit Switching
OCS	Optical Circuit Switching
BCP	Burst Control Packet
OBS	Optical Burst Switching
OP	Optical Packet
BE	Best Effort
WRON	Wavelength Routed Optical Network
D-WRON	Dynamic Wavelength Routed Optical Network
S-WRON	Static Wavelength Routed Optical Network
AWG	Array Waveguide Grating
PBS	Polarization Beam Splitter
IGP	
OSPF	Open Shortest Path First protocol
LSP	Label Switching Path
FEC	forwarding Equivalence Class
LER	Label Edge Router
LSR	Label Switching Router
RSVP-TE	Resource Reservation Protocol-Traffic Engendering
LIB	Label Information Base
CoS	Class of Service
TE	Traffic Engendering
TSC	TDM Switch Capable
LSC	Lambda Switch Capable
FSC	Fiber Switch Capable
PSC	Packet Switch Capable

OSPF-TE	**Open Shortest Path First protocol- Traffic Engendering**
LSA	***Link State Advertisement***
CSPF	**Constrained base Shortes Path First protocol**
VoIP	**Voice Over IP**
bups	**burst per second**
Gasco-tab	**Gasco- tabel**