A Framework for Optimized Bandwidth Allocation to LSPs in an MPLS Network

THESIS SUBMITTED IN PARTIAL FULFILLMENT FOR
THE DEGREE OF

Bachelor of Technology
in
Computer Science and Engineering

By
Sarah Panda
Roll no:107CS042
Nitish Shukla
Roll no:107CS045

Under the Guidance of
Prof. A. K. Turuk

Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela-769 008, Orissa, India

National Institute of Technology
Rourkela

CERTIFICATE

This is to certify that the thesis entitled **"A Framework for Optimized Bandwidth Allocation to LSPs in an MPLS Network "** submitted by **Sarah Panda : 107CS042** and **Nitish Shukla : 107CS045** in the partial fulfillment of the requirement for the degree of Bachelor of Technology in Computer Science Engineering, National Institute of Technology, Rourkela , is being carried out under my supervision.

To the best of my knowledge the matter embodied in the thesis has not been submitted to any other university/institute for the award of any degree or diploma.

Prof. A.K.Turuk
Professor and Head

date : Department of Computer Science and Engineering

National Institute of Technology

ACKNOWLEDGEMENT

We wish to express our sincere and heartfelt gratitude towards our guide **Prof.A.K.Turuk**, Computer Science Engineering Department, for his guidance, sympathy, inspiration and above all help during the duration of our project.

We would also like to thank all the professors of the department of Computer Science and Engineering, National Institute of Technology, Rourkela, for their guidance and the inspiration.

Submitted by:

Sarah Panda
Roll no-107CS042
Nitish Shukla
Roll no-107CS045
Computer Science and Engineering
National Institute of Technology
Rourkela

Abstract

Bandwidth allocation is a vital issue in the emerging MPLS technology in the area of computer networking. There is need to ensure an efficient and congestion free traffic through suitable bandwidth allocation. Though some algorithms exist to address this issue, it is felt that more optimized algorithms can be beneficial. “Weighted Max-Min congestion control algorithm” [1] by Marty and Ali, proposed a basis of congestion control. The “Weighted Proportional fair rate allocation algorithm”[2] and“Adaptive Bandwidth Allocation Algorithm”[3] addressed the issue of congestion control in MPLS networks. The above approaches used the concept of predefined weights to the LSPs which means that bandwidth is allocated according to some presumptions. This may lead to some amount of unused bandwidth and a situation may arise where bandwidth is allocated to an LSP which doesnt utilize it fully but there exists another LSP which falls short of its current bandwidth requirement. To account for the changing bandwidth needs and also the current datarate of the LSPs, this paper proposes a framework for fair bandwidth allocation to the LSPs in a more optimized manner. In addition to the algorithm, we include a simulation of a static bandwidth allocation approach using RSVP-TE with MPLS in OMNET++ IDE integrated with INET framework. We compare the parameter of queue length for all interfaces of all LSRs in the network and for a particular interface at different datarate values. We show from our observations that with increasing datarate, the average queue length gradually decreases.

Contents

List of Figures

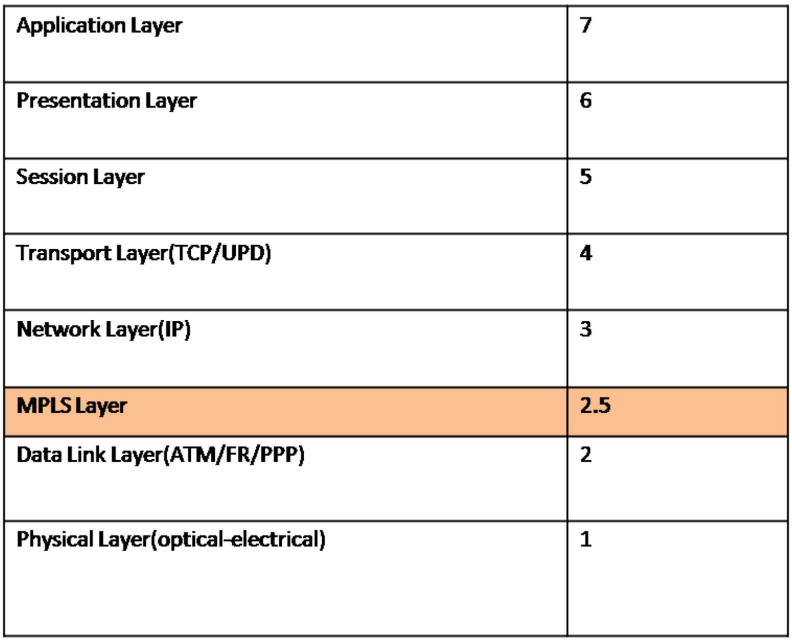
Chapter 1

INTRODUCTION

Multiprotocol Label Switching (MPLS) is an emerging technology in the area of computer networking. It has vital applications in telecommunication networks, optical switching networks and Virtual Private Networks (VPN)[11]. It makes use of labels to create virtual links for data transmission between network nodes. It is a packet switching technology having features of circuit switching due to the introduction of the virtual channels using labels. The connection oriented feature makes the transmission faster through speeding up of the address lookup during routing. Nowadays, IP backbones are made MPLS-capable to make use of this feature[11].

The packets of various network protocols are similarly treated in the MPLS networks. Every packet entering the MPLS cloud is encapsulated into an MPLS packet with an additional header containing the label. However for packets already supporting the virtual circuits, e.g. Asynchronous Transmission Mode (ATM), the label is included in the Layer-2 header. In fact, MPLS is an advancement over the earlier used networking technologies viz. ATM and Frame Relay[11]. MPLS is conceptualized based on the benefits (e.g. connection-oriented services) and weaknesses (e.g. high overhead cost) of ATM.

OSI model places MPLS layer between Data Link Layer and Network Layer[5].



Application Layer	**7**
Presentation Layer	**6**
Session Layer	**5**
Transport Layer(TCP/UPD)	**4**
Network Layer(IP)	**3**
MPLS Layer	**2.5**
Data Link Layer(ATM/FR/PPP)	**2**
Physical Layer(optical-electrical)	**1**

Figure 1.1: *MPLS Layer in OSI Model*

1.1 Structure of an MPLS Cloud

An MPLS cloud consists of various routers that support MPLS and are known as Label Switching Routers (LSRs)[5]. The LSRs which are in the periphery of the MPLS cloud are called Edge LSRs or Label Edge Routers(LERs) and must be capable of accepting packets from all types of networks. The end-to-end virtual path that is set up with the use of labels is known as LSP(Label switched path)[11]. An LSP starts at the ingress node and terminates at the egress node passing through several intermediate routers.

1.2 Forwarding Equivalence Class(FEC)

An FEC is a method for categorizing packets based on parameters like destination address, source address, TCP/UPD port, class of service or application used[5]. Depending on the FEC(Forwarding Equivalence Class) a packet belongs to, the labels are assigned. A label is a short identifier used to define a path (LSP) within an MPLS network for different FECs[5]. It can be designated by an integer or string. This type

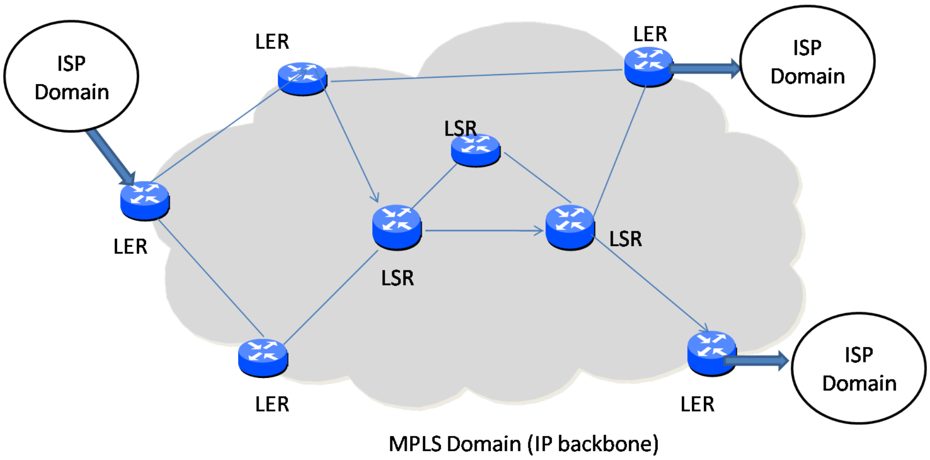


Figure 1.2: *Structure of MPLS Cloud*

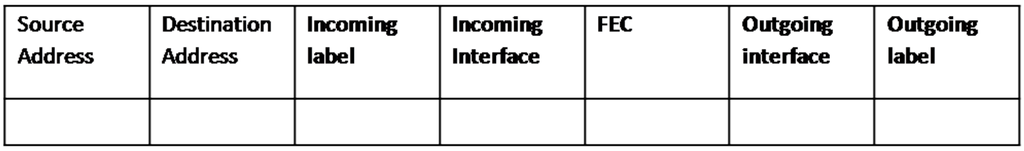
of classification makes it easy to make forwarding decisions as all packets belonging to the same FEC are forwarded on the same LSP. The assignment of labels based on the classification is done at the LERs.

For example, we can take all packets with destination address as 138.120.6/24-xxxx to belong to one FEC named‘A’. All packets destined for the above set of addresses are sent over the same LSP, designated by the outgoing label and outgoing interface in the forwarding table.

1.3 Forwarding Table of an LSR/LER

The LER forwarding table has fields viz. source address, destination address, FEC name, incoming interface, incoming label, outgoing label and outgoing interface[5]. The incoming label field in LERs may be set to NULL in cases where the packet has just entered the MPLS cloud through the LER.

This is the structure of the Forwarding table with the most important fields high-



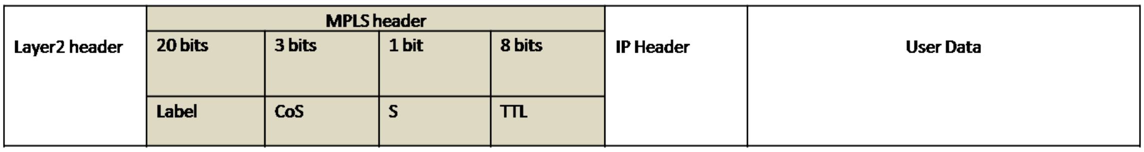
Source Address	Destination Address	Incoming label	Incoming Interface	FEC	Outgoing interface	Outgoing label

Figure 1.3: *LSR Forwarding Table*

lighted.

1.4 MPLS Header

As mentioned in [7], the structure of MPLS header in MPLS packet has the following structure.



Layer2 header	MPLS header				IP Header	User Data
	20 bits	3 bits	1 bit	8 bits		
	Label	CoS	S	TTL		

Figure 1.4: *MPLS Header format*

Label stores the actual value of the label

CoS Class of service applied to the packet which helps in deciding the priority of the packet while forwarding or discarding the packet or queuing it.

S Stack field which is set for the end of label stack[5]

TTL provides IP TTL functionality

1.5 Working of a simple MPLS network

An edge LSR inserts an MPLS label to the header of an incoming packet depending on what FEC class the packet belongs to. At every intermediate LSR for an LSP, the incoming interface and interface incoming label are matched in the forwarding table. The outgoing label in the corresponding entry then either replaces the older label in the packet header or is simply pushed into the header to form a label stack. The outgoing interface in the same entry determines which outgoing line the packet has to follow to reach its destination through the chosen LSP[5]. At the packets destination node which happens to be an LER, the label or the stack of labels are popped out and the header is removed. Here, the original packet that entered the MPLS cloud is recovered and is sent to the destination network. Hence, within the MPLS network, the original packet header is not examined. The packet remains intact within the MPLS cloud. Also, the FEC field of the forwarding table is used only at the LERs to determine the corresponding LSP. It has no function in the intermediate routers.

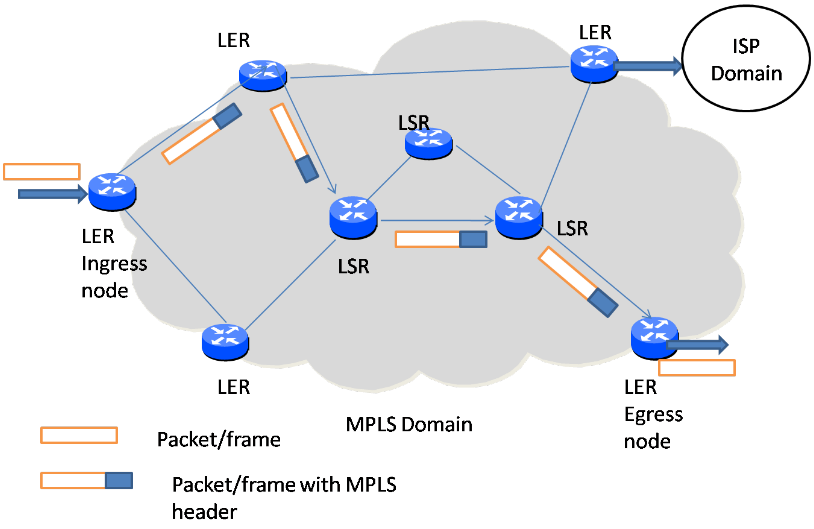


Figure 1.5: *Packet Forwarding in MPLS Network*

1.6 Weight of an LSP

Several LSPs can be used to connect a pair of network nodes. These LSPs are assigned weights which determine the priority of the LSP. Data transmission between two nodes first follows the LSP with highest weight. On failure of this LSP, the transmission follows the LSPs in the decreasing order of their weights[8].

Chapter 2

PROBLEM STATEMENT AND EARLIER APPROACHES

Here, we take up the issue of bandwidth allocation in an MPLS network. The problem statement goes as follows:
In an MPLS network, how can dynamic information about available resources be passed among routers, that will help in the allocation of traffic to the LSPs so that each node knows where the traffic must be forwarded next to avoid congestion paths. In the MPLS network, several LSPs may share the same link as shown in the following figure. LSP1 and LSP2 share the link AB, while LSP3 and LSP2 share the link CE.The capacity of the link has to be fairly distributed among the LSPs so that each of them may be used at any point of time. Theoretically, an LSP having the higher weight should be given a greater share of link capacity because it is likely to be used prior to other alternatives. A fair bandwidth allocation strategy in [1] and [2] was proposed earlier where the capacity is distributed in proportion of the weight carried by an LSP. This strategy conforms to the theoretical requirement.

The Weighted proportional fair rate allocation algorithm proposed in [2] makes use of the above mentioned fair bandwidth allocation strategy and a two-way feedback control mechanism to control the inflow of data at the ingress router. The algorithm intelligently calculates the amount of bandwidth to be permitted into the MPLS network for each LSP.
The One-way Feedback control based adaptive bandwidth allocation algorithm, in [3]

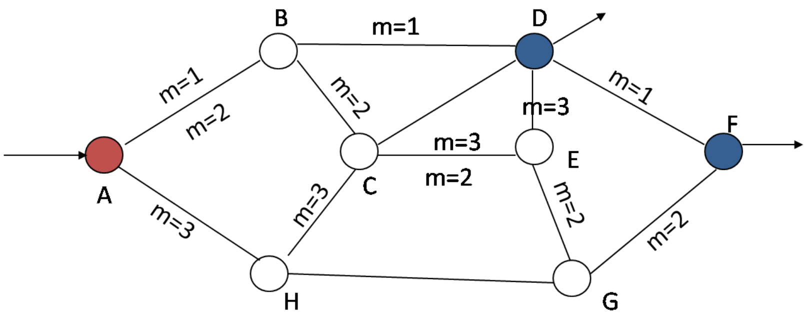


Figure 2.1: *Multiple LSPs sharing a single link*

suggests a mechanism by which the ingress router knows how much bandwidth is to be allowed to each LSP passing through it to avoid congestion. It is dynamic but will show the most effective result only when all the LSPs are in use simultaneously. When only a few of them are in use, some bandwidth may be wasted while other LSPs may require more. To avoid this, we may allow the deviation from this collision free kind of smechanism and permit more traffic if bandwidth is available. The above algorithm may be combined with the AIMD technique in MPLS as in [10] to add the benefits of the two. Here it is assumed that bandwidth allocation is made on demand and not statically.

Chapter 3

LITERATURE REVIEW

3.1 Generalized MPLS

Generalized multiprotocol label switching as described in [5], which is also called Multiprotocol Lambda switching is a multi-purpose control plane technique that supports not only packet switching but also time, wavelength and space domain switching. Generally in connection-less networks packet forwarding is performed in an independent manner at each router present in the network and relies on the destination address carried in the packet. This packet forwarding technique only supports multi-point to point path abstraction. However, recently additional functionality has been added to IP routing architecture and protocols under the umbrella of MPLS. One of the main aspects of MPLS is the addition of point to point path abstraction. This is done by the concept of label switched paths(LSP).The connectivity abstraction supports constraint based routing which in turn is the basis for Generalised Multiprotocol Label Switching[5].

One of the applications of MPLS is constraint based routing, which is often used to compute paths that satisfy certain requirements to a set of constraints. Constraint based routing is used for Traffic Engineering and Fast Reroute. MPLS constraint-based routing allows nodes to exchange information not only about network topology, but also about availability of resources and administrative constraints. This information is used as the input to any constraint-based path computation program that computes paths on the above mentioned parameters. After finding an appropriate path signaling protocols such as Resource Reservation Protocol with Traffic Engineering (RSVP-TE) is used to initiate a label forwarding system across the path. Recent improvements have been done to enable using MPLS constraint based routing in optical cross connects. This is an important step in the integration of optical network and data architectures. Use of MPLS as the basis for establishment of connections and a common control plane helps addressing several issues in network evolution. Firstly, network operations and management are simplified by using a common control plane, which ultimately reduces operational costs. Secondly, a common control plane provides a huge range of deployment scenarios. This allows us to choose the peer or overlay deployment models to be based and modeled on business and engineering considerations, instead of being restricted by stratification of sub-networks into technology domains. Also, development of a common control plane minimizes the risks generated with protocol development and reduces the time needed to market for enhanced optical switching equipment[5].

Some additional features have been added to GMPLS to manage some of the drawbacks in MPLS control plane. These include inability to manage connection in a bidirectional manner in one request and lack of mechanisms to protect bandwidth which could be used for low-priority traffic. In the MPLS framework a link or node failure could only be handled locally or across the nodes of the path, however in the GMPLS framework additional functionalities, such as ability to report to a predefined alarm centre in case of a failure which impacts service connections, have been added.

Enhancements to Signaling

In GMPLS, we need similar devices as start and end points of LSPs. MPLS is designed to ensure that the data plane is logically distinct from the control plane. GMPLS extends this to incorporate the data plane being physically distinct from the control plane. GMPLS is an example of a scalable, generalized, and manageable architecture.

Hierarchical LSP Setup

GMPLS uses the concept of hierarchical LSPs. This uses the concept of tunneling. Here, a new LSP is tunneled inside a pre-existing higher order LSP, such that the pre-existing link acts as a link along the new LSP path. Hence, lower order LSPs often trigger the formation of higher order ones. The responsibility for creating higher order LSPs and aggregating lower order ones is on the nodes at the border of the two regions involved.

Bi-direction LSP Setup

Many optical networking service providers require bidirectional LSPs. GMPLS supports bi-directional LSPs. It is taken for granted that both sides have the same traffic engineering requirments. There is an initiator and terminator node. The initiator node refers to the source node and the terminator node refers to the destination node. In GMPLS there is only one initiator and terminator node. In MPLS , since we can only set up unidirectional connections, to set up a virtual bidirectional connection, two unidirectional connections are set up in opposite directions. Thus, there are two initiator and terminator nodes each. This method however has a lot of disadvantages compared to the bidirectional LSP concept in the GMPLS framework paradigm.

GMPLS will be an integral part of the next generation of optical and data networks. It forms the important link between IP and photonic layer. The functionality provided by GMPLS allows the operators to scale applications well beyond current limitations in the network field. GMPLS provides signaling capabilities which will allow providers to build high capacity architecture which will allow fast provisioning of connection services. Also, the restoration capabilities of GMPLS will enable efficient addressing of network survivability.

3.2 Creating Label Switched Paths in MPLS network

3.2.1 Constraint Based Routing

Constraint based routing [4] finds paths which are subject to various constraints such as bandwidth allocation and administrative policies. Since it considers more than just network topology while finding a path, constraint based routing might find a longer but less loaded path as compared to a shortest path which is heavily loaded. Network traffic is therefore distributed more evenly.
Consider the example given below. Here, the shortest path exists between routers A and B. However, since the reservable bandwidth on the shortest path is only 35 Mbps for an LSP of 50 Mbps, we select the Router A-B-C path.

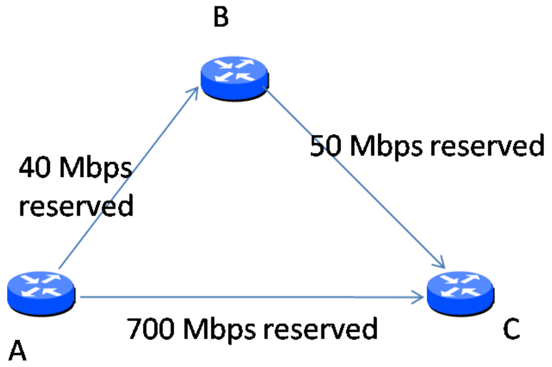


Figure 3.1: *Constraint Based Routing*

Constraint based routing can be of two types - offline mode and online mode. In the online mode routers may compute paths at any instant of time while in the offline mode routers compute paths only periodically.

3.2.2 Enhanced Link State IGPs

For the constraint based routing to be able to compute LSP paths based on constraints, an enhanced link state IGP as given in [4] can be used to send link attributes along with the usual link state information. An example of a link attribute is reservable bandwidth.

Compared to a normal IGP, an enhanced link state IGP floods information at more frequent intervals. This is because a normal IGP floods information only when there is a change in topology. However, even without any change in topology, an enhanced link state IGP floods information due to change in link attributes such as reservable bandwidth. However there should be a trade off to prevent excessive flooding. Thus only when there is a significant change in bandwidth(above a certain predefined threshold) flooding occurs.

The enhanced IGP then builds the LSRs forwarding table[4].

3.3 Bandwidth allocation in MPLS networks

Different bandwidth allocation strategies are categorized as follows:

3.3.1 STATIC METHODS

a) The crudest way to divide link capacity between LSPs is to distribute them equally. The bandwidth hence allocated is simply the maximum reservable bandwidth for a link divided by the number of LSPs passing through it which means equal allocation of bandwidth to LSPs irrespective of their requirement i.e. if 3 LSPs pass through a link with maximum capacity of 90Mbps, each LSP passing through it is given a share of 30 Mbps. High chances of wastage or under-utilization of bandwidth exist with this approach .
It is simple but does not take into consideration the priority of the LSPs. Hence, it is an unfair allocation of bandwidth. b) A better approach is to divide the link capacity by taking a weighted average. So an LSP gets a fraction of the capacity in proportion to the bandwidth it requests for. This is also known as fair bandwidth allocation [2]. Let multiple LSPs pass through a link with link capacity C. LSP I requests for bandwidth w(i).
So according to the fair bandwidth allocation,

$$R(i) = w(i) * C/(w(1)+w(2)+.+w(n))$$

Where R(i) is called the optimum bandwidth for LSP I and n is the number of LSPs passing through the specified link[2].

The static methods are not adaptable to changes in network configurations or network traffic. The ever increasing traffic and users poses a requirement to optimize the limited resources as per current usage and other factors.

3.3.2 DYNAMIC METHODS

Several algorithms have been proposed to address the fairness issue with dynamism taken into account. The one-way feedback control based congestion control algorithm

[3] proposed earlier is based on the Weighted Proportional Fair Rate Allocation Algorithm (WPFRAA) proposed in [2]. In this section, we summarize the working of the WPFRA algorithm and then the one-way feedback based algorithm. Our framework is based on the these two approaches clubbed with an approach similar to that of AIMD(Additive Increase/Multiplicative Decrease) used for congestion control [10]. It is described in the next section.

a) Weighted Proportional Fair Rate Allocation Mechanism

This mechanism as proposed in [2] involves four different functions viz. measurement, calculation, notification, enforcement. Each of these four phases is described here.
Measurement is carried out at the core routers. The traffic at a core router is measured here. This aids to the calculation phase of the mechanism.
Next comes the **calculation** phase. The quantity calculated here is the optimum bandwidth allocation to each LSP passing through the router in the forward direction i.e. in the direction of ingress node to the egress router. Each of the LSPs in the network is assigned a weight value. The following equation is used for the calculation:

$$r_i = w_i . \frac{C}{\sum_{j=1}^{n} w_j}$$

Where C is the link capacity, i is the LSP index unique to each LSP and n is the total number of ‘active’ LSPs passing through the link. Traffic analysis is done to know the status of traffic flow through the network and hence calculate the value of n. This value of ‘ri’ is then operated with the actual traffic flow and a smoothening factor to get ‘rf’.
Notification is the next phase. The ingress node sends out a control packet along each LSP originating at it. The control packet includes an Explicit Rate(ER) field [1] that is initialized to a very high value. When the control packet reaches a core router, the rf value is compared with the existing ER in the packet. The minimum of the two values becomes the new ER of the packet which is then sent out on the same LSP. So at each step, the minimum value of the available bandwidth is written into

the ER field of the packet. When the packet reaches the egress router it is returned back to the ingress node without any further modifications along the path.

The last of all phases is **enforcement**The ingress router now has the explicit rate values of each LSP passing through it. A leaky-bucket algorithm is implemented at the ingress node whereby the amount of traffic entering the MPLS network is restricted to the corresponding ER values. This avoids any sort of congestion along the path, which could have occurred if more bandwidth would have been allowed in and couldnt be handled by the network.

For the notification to reach the ingress router, the control packet has to make a round trip around the MPLS network. The overhead involved in the round trip is substantially reduced in the one-way feedback based congestion control mechanism proposed in [3].

b) ADAPTIVE BANDWIDTH ALLOCATION

This algorithm [3] is based on the WPFRA mechanism given in [2] with a modification that the round trip of the control packet in the notification process is replaced by a one- way feedback control. It is used to lessen the convergence time for bandwidth allocation. It bases itself on the fact that the path in an LSP in MPLS is reversible because of the use of labels. As an incoming label is mapped onto an outgoing label through the information in the forwarding table, the reverse mapping is also possible because of the one-to-one correspondence between the labels. The reverse mapping is stored in a table known as Inverse Label Mapping (ILM) table. This way an LSP can be retraced starting at the egress node in the reverse direction.

The working of Adaptive Bandwidth Allocation is summarized in the following steps:

1. At an interval, known as Measurement Interval(MI), the core routers measure the instantaneous traffic and calculate the optimum bandwidth.
2. Another interval, known as Notification Interval(NI), triggers the egress router for every LSP to send a control packet along the reverse path with ER field initialized.
3. At each core router along the path, the ER value is updated as in WPFRA mechanism
4. The final ER values are with each edge router. A leaky bucket algorithm is used to filter traffic with the ER values set as policing parameters.

Chapter 4

A FRAMEWORK FOR OPTIMIZED BANDWIDTH ALLOCATION

There are a few terms which need to be defined before we start with the framework. They include weight of an LSP, priority of data traffic and status of an LSP.

Weight as explained before, is the priority assigned to the LSPs between two nodes as given in [8]. A route with higher weight is preferred over other routes. The priority assigned to the LSPs is the bandwidth it requests from the MPLS network. If between two pairs of network nodes there are several LSPs, the one with more requested bandwidth has a greater priority and is assigned more bandwidth compared to others during any fair bandwidth allocation strategy.

Priority of data traffic indicates the type of data being transmitted. Data with higher priority is not allowed to compromise on the bandwidth allocated to it. It will be elaborated later.

We consider three types of **status** of an LSP: up, down and active[8]. An 'up' route is one which is capable of carrying traffic but is not being used currently possibly because some other route with higher weight is available. An 'active' route is one which is currently being used for data transmission. An active path is by default an 'up' route. But an up route may not be active. We consider this difference in our framework. A 'down' LSP is one that is unable to carry traffic for some reason but has been defined in the forwarding tables.

In the previous approaches, the optimum bandwidths were calculated for the 'active' LSPs. Our framework takes into account all 'up' LSPs for optimum bandwidth calculation.

4.1 A proposal for effective bandwidth allocation

Our project aims at finding a new technique or improving an existing technique for efficient dynamic bandwidth allocation and efficient notification mechanisms. So to start with, we took reference of the above algorithm and identified the factor that could restrict the network performance when it was used. We, then, tried to find a tentative solution to the issue which is presented here. These type of algorithms fall under the category of OPEN LOOP Problems that include deciding when to accept new traffic, deciding when to discard packets and which ones, and making scheduling decisions at various points in the network.

We describe the scenario using a problem as below.
We assume the network as given below.

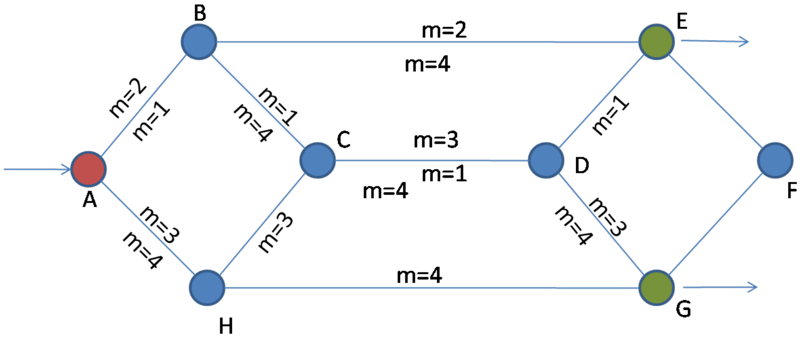


Figure 4.1: *An MPLS network with four LSPs defined.*

The following table gives the values of some parameters for the four LSPs in the network.

LSP(m)	Current Data Rate	Weight
1	10	20
2	20	30
3	15	30
4	10	15

The value for current data rate is determined by observing the data traffic in the LSPs over an interval and finding the average data rate through the path.

Let capacity of link CD be 60.

Case:

m=1,3 : active, m=4 : up
By WFPRAA, the share of capacity of link CD reserved for m=1 i.e. the optimum bandwidth of LSP1 (OB at CD,1)

OB at CD,1 = 60 * 20/(20+30) =24
OB at CD,3 = 60 * 30/(20+30) =36

By adaptive bandwidth allocation algorithm, let the ER values be
ER(LSP1) = 20
ER(LSP3) = 32
Free_bandwidth(CD)=8

This is the amount of capacity of link CD reserved for LSP1 and LSP3 respectively. The rest is free to be reserved by other LSPs demanding bandwidth.
As we can see, the current data rate for the LSPs is lower than the allocated bandwidth. So part of the allocated bandwidth is unused. It is wise to free the unused bandwidth to be used by other LSPs wanting to be active or requesting more bandwidth. So, we reduce the bandwidth allocated from the ER values to the current data rate values. The rest is added to the Free_bandwidth parameter of the link.
So, bandwidth allocated to LSP1=10
and to LSP2 = 15
Free_bandwidth(CD)=35

Now, for any new LSP, bandwidth can be allocated from the above Free_bandwidth(CD).

Chapter 5

ALGORITHM IN DETAIL

There are a few terms we consider for an LSP.

Weight : Average requested bandwidth of an LSP as in [1]

Requested bandwidth : Instantaneously requested bandwidth.

Current Data Rate : the actual transmission rate measured over an interval as defined in [1].

This algorithm is based on the fact that the current data rate could be lesser or more than the allocated bandwidth. Bandwidth is allocated through the ER values, which depend on the weight of the LSP which is an average value and may not always indicate the actual values. This algorithm makes up for the flaws in the assumption of the weight.

The steps followed:

1. Allocate bandwidth as per the WPFRA algorithm which results in a fair bandwidth allocation to all active LSPs based on the weights associated with them.
2. Current Data Rate of LSPs is calculated and compared with the allocated ones. Accordingly the allocation is adjusted to free any unused bandwidth. This freed bandwidth is added to a parameter, free_bandwidth and least_free_bandwidth of each node and each ingress node for an LSP, respectively.
3. At intervals, a function called reset_free_bandwidth makes up for any changes to the links of the LSP.

4. The free_bandwidth is used to allocate more bandwidth to the LSPs wanting to become active or whose bandwidth allocation is lesser that the requested bandwidth.

Here, we cover a few cases which may be merged together in their description:
- When a new LSP becomes active and requests bandwidth
- When an LSP gets deactivated
- When the current data rate is lesser/more than the allocated bandwidth.
- When changes occur at Notification Interval
- When the requested bandwidth is more than the allocated bandwidth.

We also cover some issues like,
- How often should the nodes be notified.
- How to notify every ingress node.
- What information is to be passed on to the ingress router.

5.1 Changes to Control Packet

Along with the ER values, each core router has information about the free bandwidth available with each of its outgoing paths. The control packet carries information about the minimum of free bandwidths that are available along all links through an LSP which is the maximum free bandwidth that can be allocated to new LSPs.

5.2 CASE : New LSP becomes active

```
If free_bandwidth <weight
{
Rest allocations remain same
ER_new= weight
Reset_free_bandwidth()
}
```

```
Else if free_bandwidth <weight
{
If Calculated_ER >free_bandwidth
{
Reallocate bandwidth as per the newly calculated ER values
Reset_free_bandwidth()
}

    Else
{
ER_new= free_bandwidth
Rest allocations remain same
Reset_free_bandwidth()
}
}
```

5.3 CASE : An LSP gets deactivated

Allocated bandwidth of an LSP is freed. The same quantity is added to the free b/w for the corresponding links passing through the LSP. A packet initiates at the LSPs egress router whose task is to add to the actual free bandwidth along the path in the reverse direction. Reset_free_bandwidth is then called to make changes to all LSPs sharing any link with this LSP.

Let us take two LSPs sharing a link AB. If LSP1 deactivates, a message is sent in the forward direction with a purpose to notify all nodes regarding the change. Message starts at the ingress of LSP1, adding free bandwidth to the actual_free_bandwidth parameter for link 1-A. Since this is going to be deactivated, the least_free_bandwidth of the LSP parameter at 1 is set to (its previous value + the freed bandwidth)
As message reaches ‘A’, and the above modifications are made, changes are notified to the LSP2 nodes because they share the link A-B. The new message for notification starts in the forward direction of LSP so that, it reaches the egress node and waits

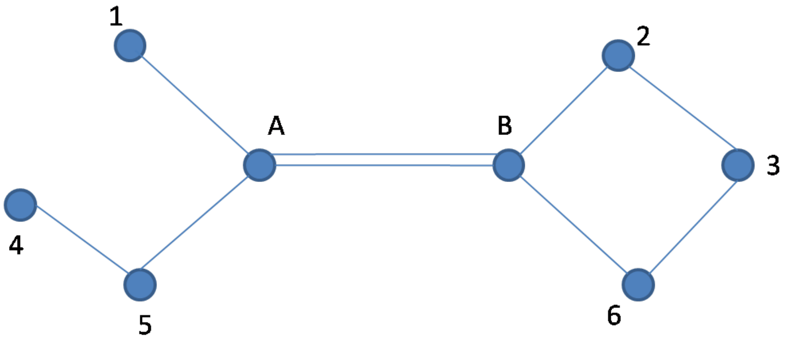


Figure 5.1: *A network of two LSPs sharing a link to demonstrate LSP deactivation*

till the deactivation information is distributed to all nodes. After this is done, the Reset_free_bandwidth starts for all LSPs whose egress router got the message.

At the end of the interval, every LSP and node which had a potential change are updated of the new free bandwidth.

5.4 CASE : Changing bandwidth allocation

Activation and deactivation of LSPs are not too frequent. But there can be frequent variations in the current data rate of an LSP.

Reallocation Measurement Interval (RMI): Current rate for each LSP is measured at the ingress node over an interval.

Reallocation Notification Interval (RNI):

```
if (allocated_bandwidth >current rate)
{
reduce allocated bandwidth to (current rate + threshold)
new_free_bandwidth = free_bandwidth + reduced bandwidth allocated
}
```

5.5 FUNCTION : Reset_free_bandwidth()

At ingress node for one of the LSPs:

After receiving the message and the new free size for the link, it is compared with the (free bandwidth) ER value.

Every router has two values for free_bandwidth of LSPi passing through it :
- The actual_free_bandwidth of the link
- The least of all free bandwidths of all the link upto that link in the backward direction of the LSP.

Now, an egress router initiates the message with a parameter as the new_free_bandwidth. The message starts from that point and traverses in the backward direction of the LSPs passing through it. At each intermediate router, now, it compares this new_free_bandwidth with the actual free bandwidth of the links and updates if necessary. This way it reaches the ingress node, with the new value of the free_bandwidth.

Chapter 6

SIMULATION

We include the simulation of a static bandwidth allocation using RSVP-TE protocol for bandwidth reservation and MPLS using OMNET++ 4.1 IDE integrated with INET framework. The network simulator OMNET++ 4.1 by itself doesnt support MPLS networking. The additional package of INET is needed to simulate MPLS networks [9].

The steps followed in the simulation as mentioned in [9]:
1. We create a new OMNET++ project in the desired folder
2. In the project–properties bar, we select the references tab and check the 'inet' box to specify that the project will use the functions of the inet package.
3. Then we create an NED(Network Definition file) under that folder to define the structure of the network using GUI tools in the framework. The NED file in the example network RSVPTE4.ned given below which we have taken from the folder inet/examples/mpls/testte_tunnel for analysis looks like the following.

Each LSR is a RSVP LSR which includes the modules rsvp, linkStateRouting, network layer, mpls, ppp, routing table, interface table, libTable etc.
Each host is a standard host with an IP address and is capable of transmitting and receiving packets.
4. We create an 'rt' file for each of the host and LSR in the network which represents the routing table information for the respective nodes. For the example network, we have seven 'rt' files defined for LSRs and five for standard hosts. To understand the

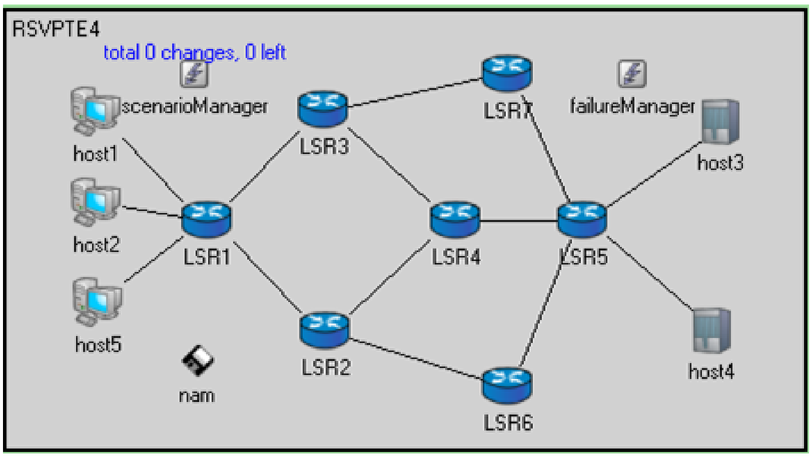


Figure 6.1: *Snapshot of RSVETE4.ned*

results and analysis given in the later chapters, we need to show the structure of 'rt' files.

host1.rt:

```
ifconfig:
name: ppp0  inet_addr: 10.3.1.1     MTU: 1500    Metric: 1
ifconfigend.
route:
10.1.1.1     *                      255.255.255.255         H            0            ppp
default:     10.1.1.1     0.0.0.0                 G            0            ppp0
routeend.
```

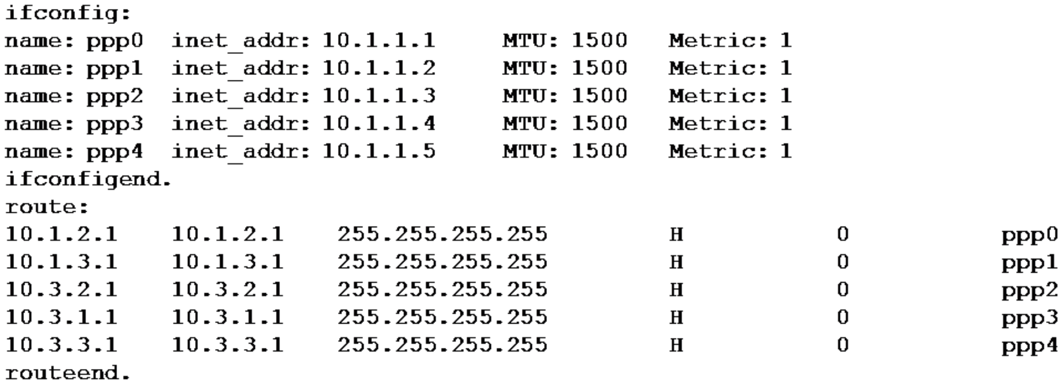
host2.rt

```
ifconfig:
name: ppp0  inet_addr: 10.3.2.1     MTU: 1500    Metric: 1
ifconfigend.
route:
10.1.1.1     *                      255.255.255.255         H            0            ppp0
default:     10.1.1.1     0.0.0.0                 G            0            ppp0
routeend.
```

Each host has one interface, defined by ppp0 and an inet_addr. The rt files are similarly defined for the rest of the network hosts.

The inet_addr for
Host3: 10.4.1.1
Host4: 10.4.2.1
Host5: 10.3.3.1

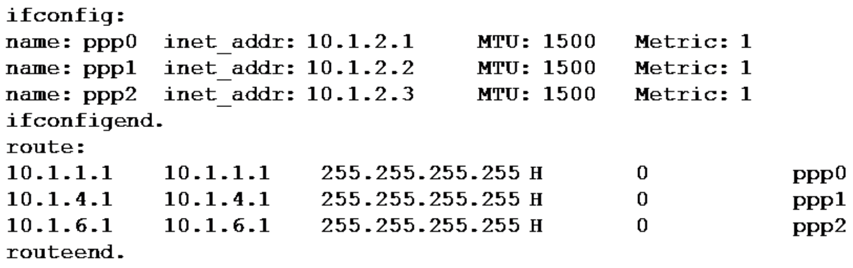
LSR1.rt:



```
ifconfig:
name: ppp0  inet_addr: 10.1.1.1     MTU: 1500   Metric: 1
name: ppp1  inet_addr: 10.1.1.2     MTU: 1500   Metric: 1
name: ppp2  inet_addr: 10.1.1.3     MTU: 1500   Metric: 1
name: ppp3  inet_addr: 10.1.1.4     MTU: 1500   Metric: 1
name: ppp4  inet_addr: 10.1.1.5     MTU: 1500   Metric: 1
ifconfigend.
route:
10.1.2.1    10.1.2.1    255.255.255.255         H          0          ppp0
10.1.3.1    10.1.3.1    255.255.255.255         H          0          ppp1
10.3.2.1    10.3.2.1    255.255.255.255         H          0          ppp2
10.3.1.1    10.3.1.1    255.255.255.255         H          0          ppp3
10.3.3.1    10.3.3.1    255.255.255.255         H          0          ppp4
routeend.
```

Since there are five interfaces, five inet_addresses are defined. The inet_address for the router would be 10.1.1.* . Routes are defined for each of its neighbor and the interfaces they refer to.

LSR2.rt:



```
ifconfig:
name: ppp0  inet_addr: 10.1.2.1     MTU: 1500   Metric: 1
name: ppp1  inet_addr: 10.1.2.2     MTU: 1500   Metric: 1
name: ppp2  inet_addr: 10.1.2.3     MTU: 1500   Metric: 1
ifconfigend.
route:
10.1.1.1    10.1.1.1    255.255.255.255 H       0          ppp0
10.1.4.1    10.1.4.1    255.255.255.255 H       0          ppp1
10.1.6.1    10.1.6.1    255.255.255.255 H       0          ppp2
routeend.
```

LSR2 is connected directly to three network nodes through interfaces ppp0, ppp1 and ppp2 and can be denoted by inet_addr, 10.1.2.*.

Similarly, other network nodes are defined in the rt files.

The inet_addr for,

LSR3: 10.1.3.* , 3 interfaces ppp0,ppp1,ppp2

LSR4: 10.1.4.* , 3 interfaces ppp0,ppp1,ppp2

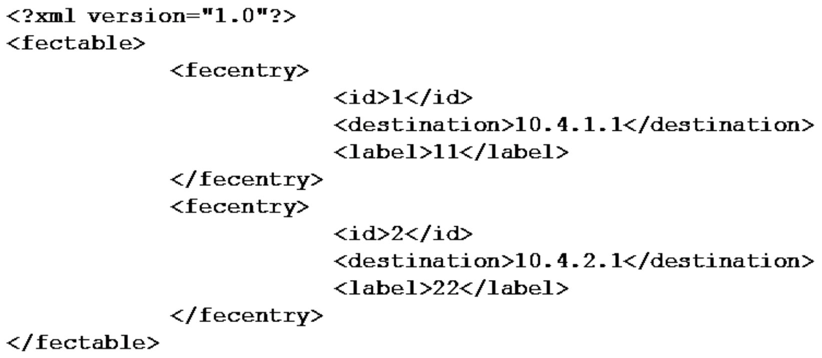
LSR5: 10.1.5.* , 5 interfaces ppp0,ppp1,ppp2,ppp3,ppp4

LSR6: 10.1.6.* , 2 interfaces ppp0,ppp1

LSR7: 10.1.7.* , 2 interfaces ppp0,ppp1

5. We define an xml file for each LSR that describes the paths from one host to the other as described below.

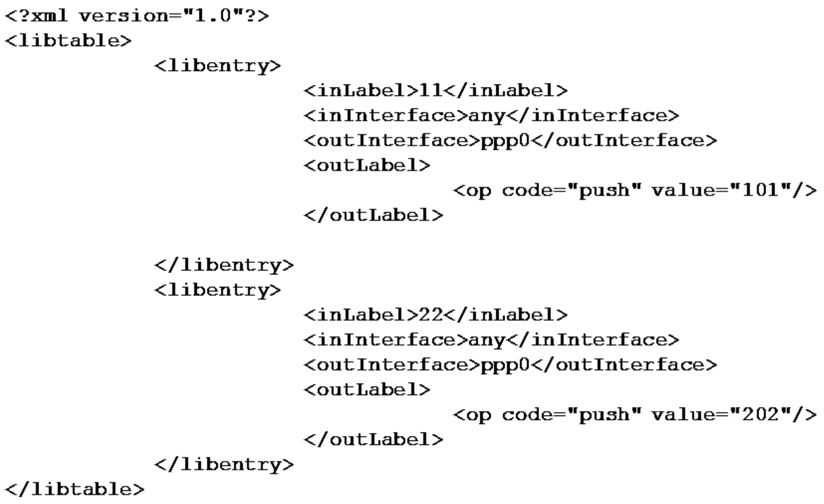
LSR1_fec.xml:



```
<?xml version="1.0"?>
<fectable>
            <fecentry>
                        <id>1</id>
                        <destination>10.4.1.1</destination>
                        <label>11</label>
            </fecentry>
            <fecentry>
                        <id>2</id>
                        <destination>10.4.2.1</destination>
                        <label>22</label>
            </fecentry>
</fectable>
```

There are two FECs defined here for packets moving out of LSR1. For all packets destined for host3, the FEC id is 1 and label inserted is 11. For packets destined for host4, the path followed is defined by FEC id, 2 and an outlabel of 22.

LSR1_lib.xml:



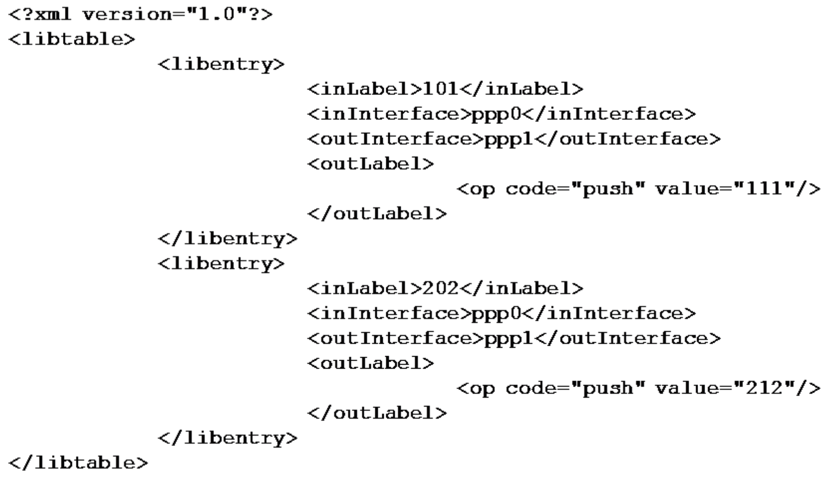
```
<?xml version="1.0"?>
<libtable>
            <libentry>
                        <inLabel>11</inLabel>
                        <inInterface>any</inInterface>
                        <outInterface>ppp0</outInterface>
                        <outLabel>
                                    <op code="push" value="101"/>
                        </outLabel>

            </libentry>
            <libentry>
                        <inLabel>22</inLabel>
                        <inInterface>any</inInterface>
                        <outInterface>ppp0</outInterface>
                        <outLabel>
                                    <op code="push" value="202"/>
                        </outLabel>
            </libentry>
</libtable>
```

This file defines the parameters for the LibTable. Parameters are read from this file and set in the corresponding Forwarding Table entry. The first libentry says that "if a packet comes with inlabel as 1 from any interface, send that packet onto the path represented by ppp0 after pushing a label 101 into the label stack"

When this encapsulated MPLS packet reaches LSR2(connected to ppp0 of LSR1) with LabelStack as 11:101, LSR2s forwarding table is searched for the corresponding entry.

LSR2_lib.xml:



```
<?xml version="1.0"?>
<libtable>
            <libentry>
                        <inLabel>101</inLabel>
                        <inInterface>ppp0</inInterface>
                        <outInterface>ppp1</outInterface>
                        <outLabel>
                                    <op code="push" value="111"/>
                        </outLabel>
            </libentry>
            <libentry>
                        <inLabel>202</inLabel>
                        <inInterface>ppp0</inInterface>
                        <outInterface>ppp1</outInterface>
                        <outLabel>
                                    <op code="push" value="212"/>
                        </outLabel>
            </libentry>
</libtable>
```

So, with topLabel 101 and inInterface of ppp0 as satisfied by the above packet, the outgoing Label stack becomes 11:101:111 as 111 is pushed into it and the packet along with this modified header is sent over interface ppp1, which is LSR4.
A packet destined for host3, therefore, follows the route LSR1, LSR2, LSR4, as defined by the xml files. This is particularly part of an LSP from any of the hosts at the left end to host3.
6. Simulation can then be run using the omnetpp.ini file.
For analysis,
7. We generate an analysis file under that project. The simulation results in some vector(.vec) and scalar(.sca) files which are loaded into the analysis file. Each of the vectors and scalars can be opened to see the corresponding graphs.

Chapter 7

SIMULATION RESULTS

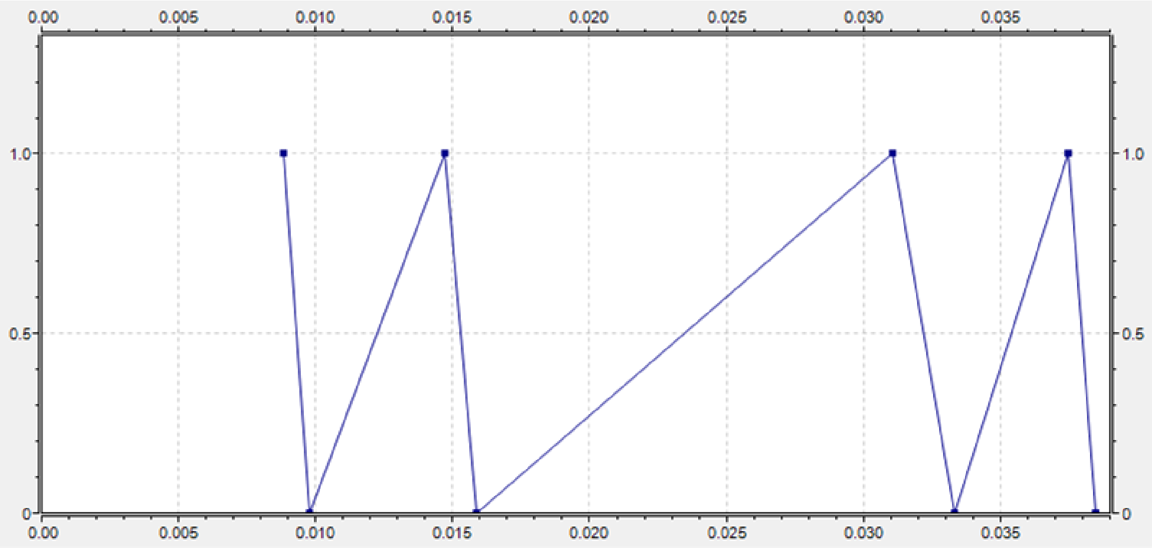
CASE 1:

Queue length for every interface of every LSR/LER in the MPLS network

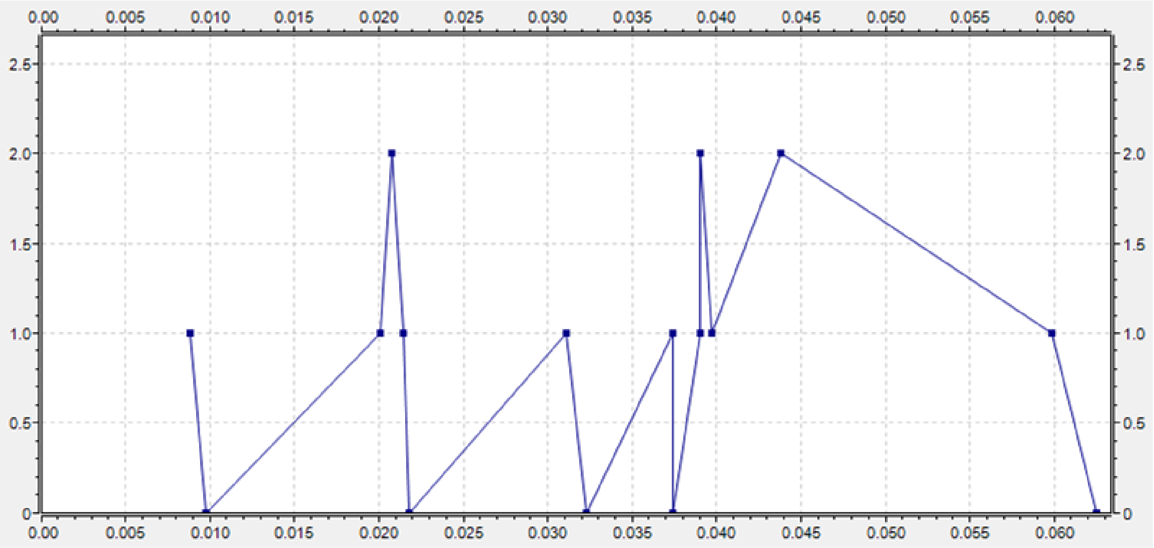
X-axis: time elapsed as an unit defined in the OMNET++ simulation environment (0.001 units = 1 sec)

Y-axis: number of packets being queued at the interface at that time

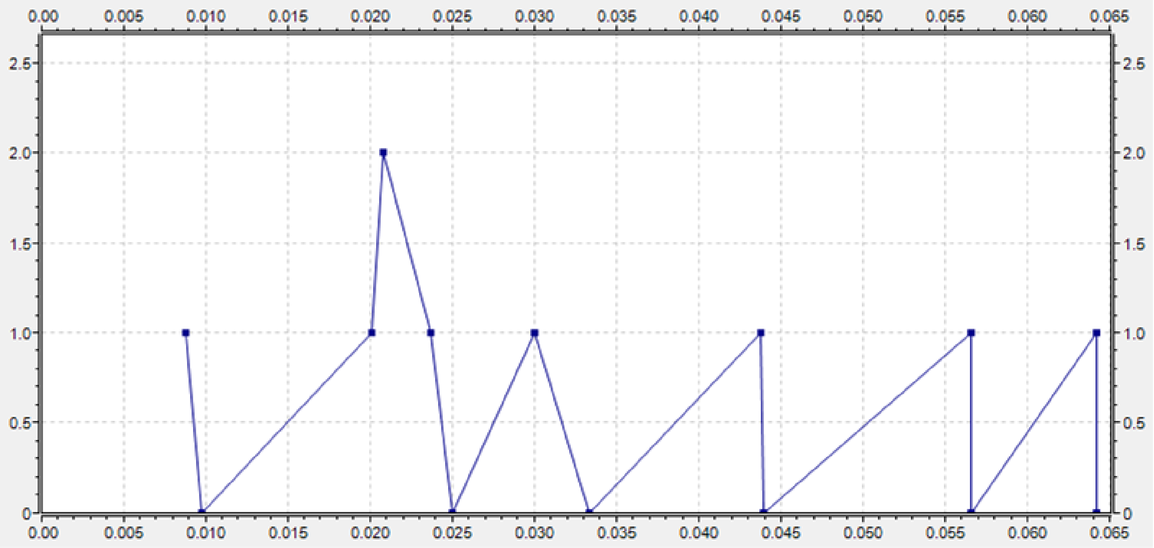
Graph 1: LSR4.ppp[0], mean=0.5



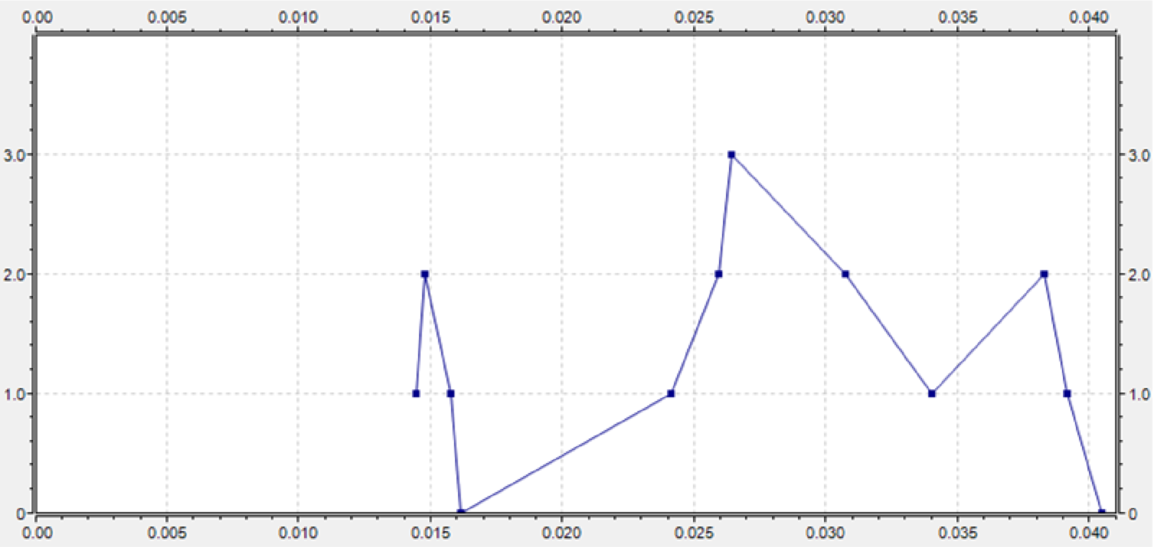
Graph 2: LSR4.ppp[1], mean=0.875



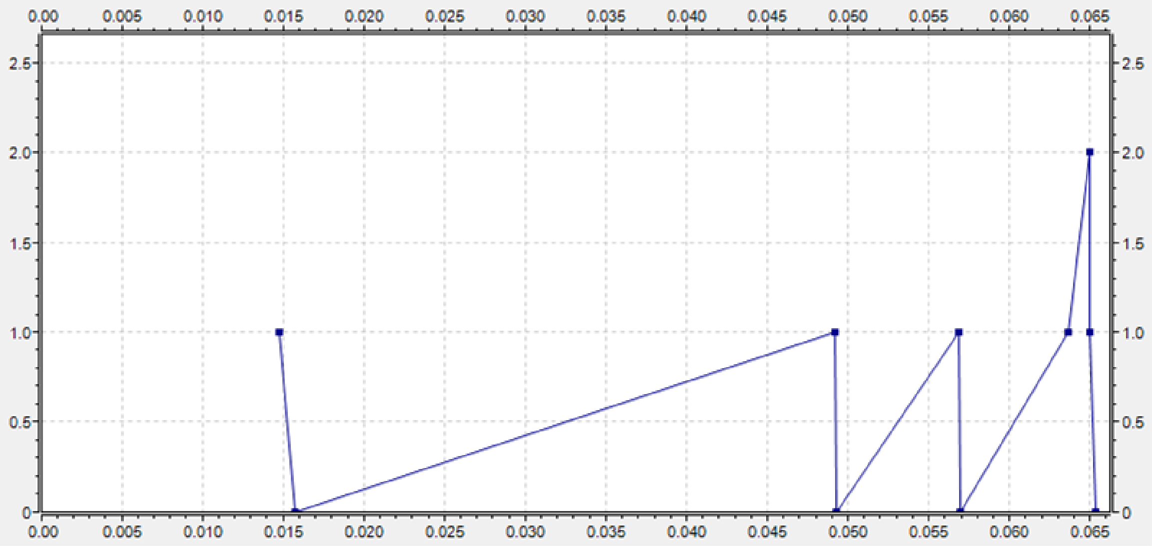
Graph 3: LSR4.ppp[2], mean= 0.6428



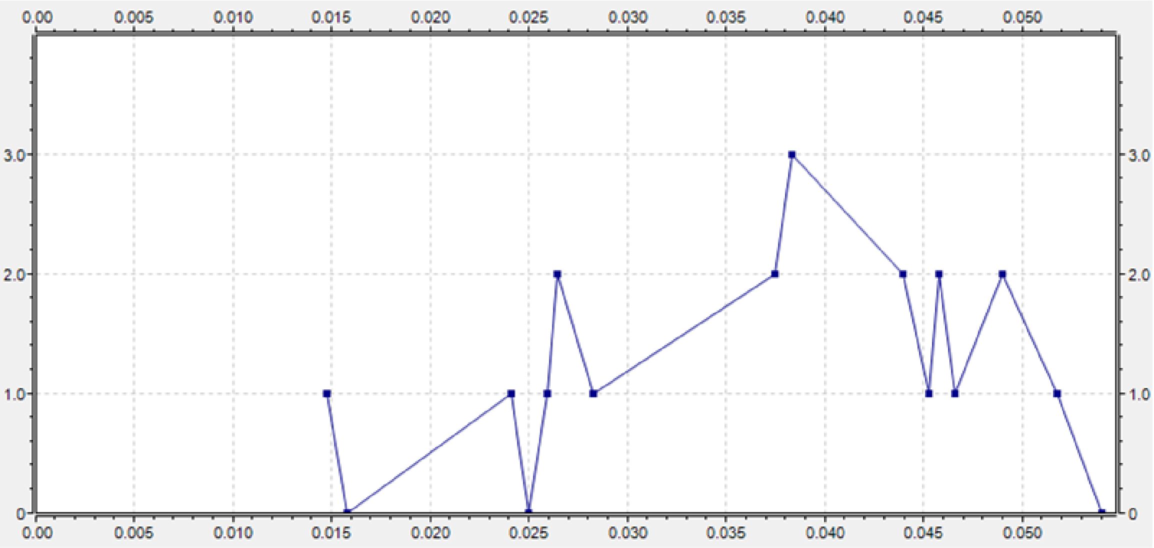
Graph 4: LSR2.ppp[2], mean=1.3333



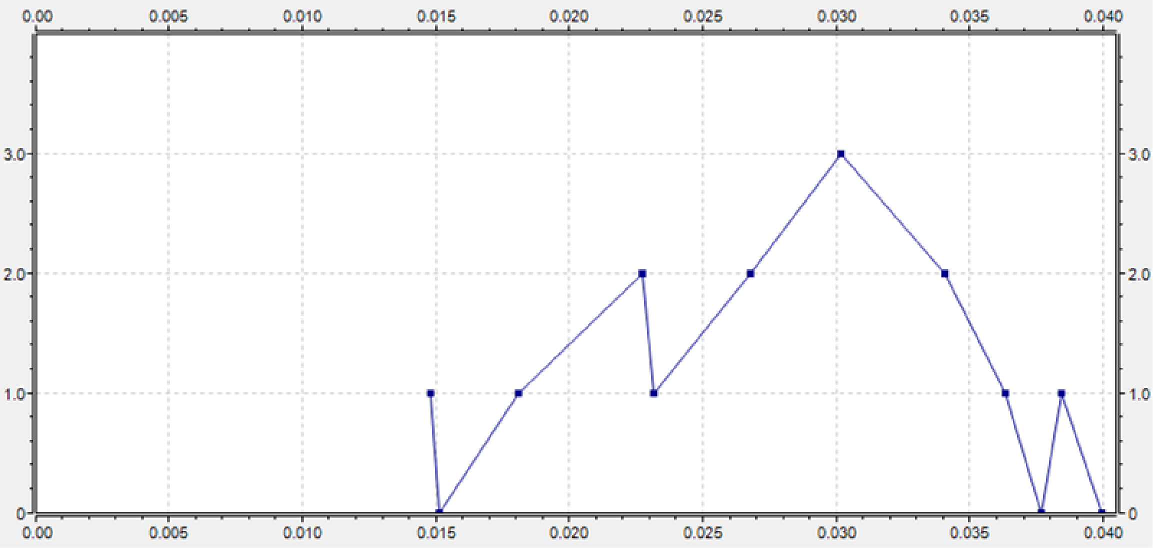
Graph 5: LSR2.ppp[1], mean=0.7



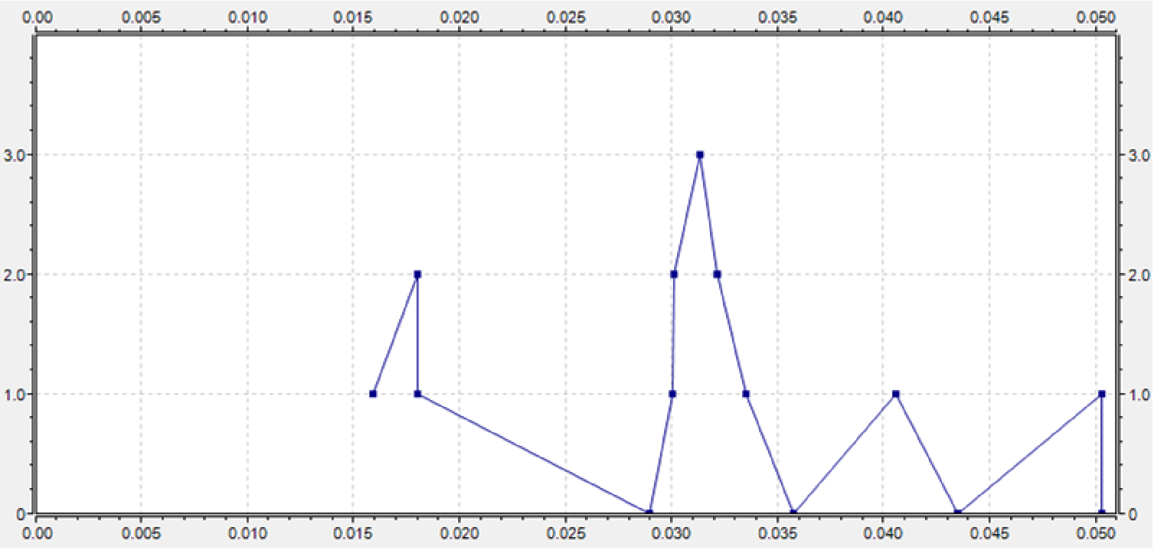
Graph 6: LSR2.ppp[0], mean=1.25



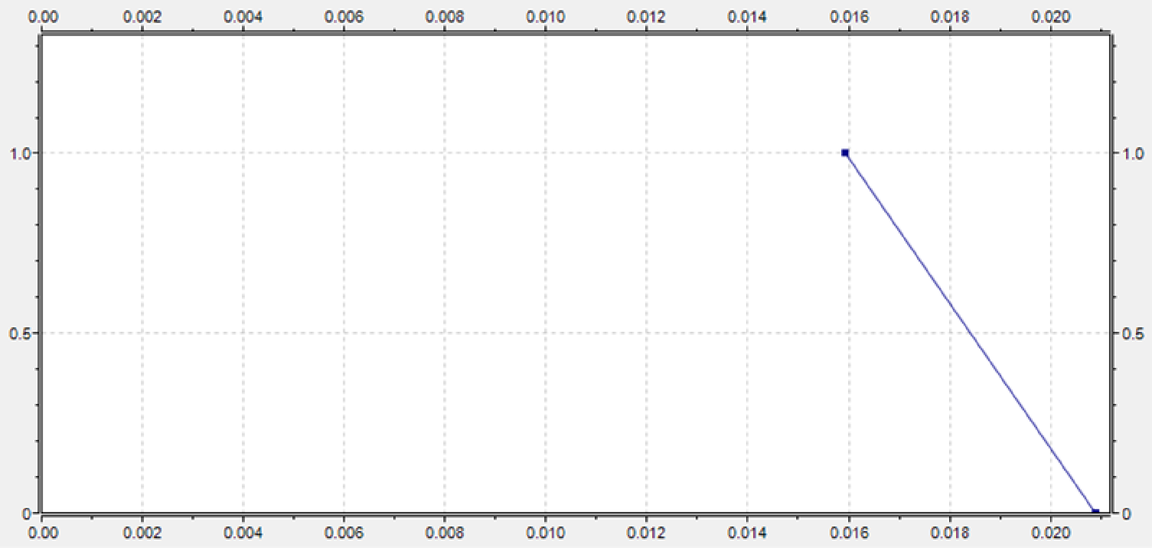
Graph 7: LSR5.ppp0, mean =1.1666



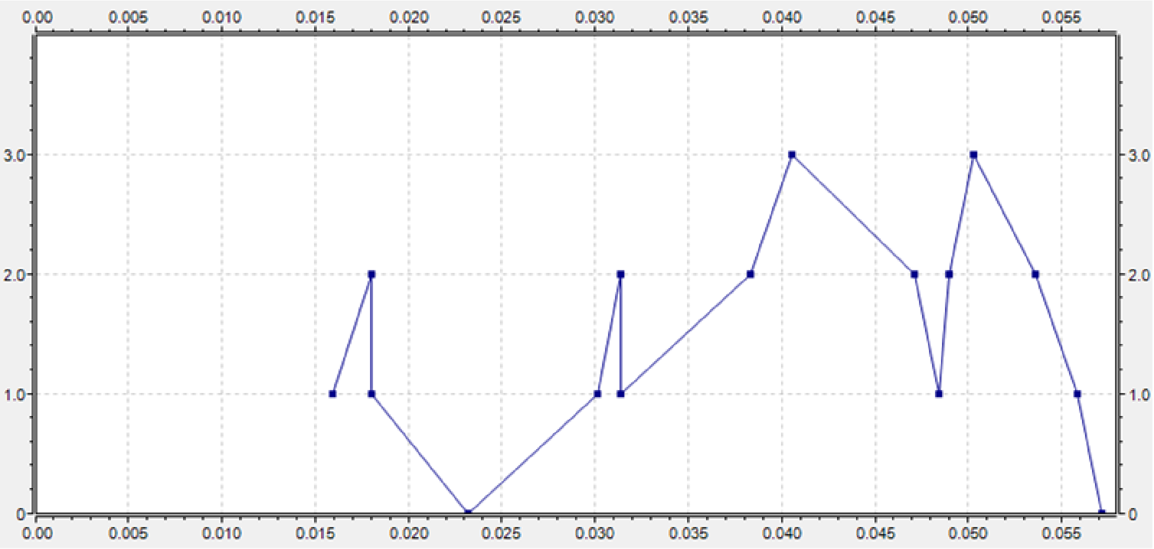
Graph 8: LSR3.ppp0, mean=1.0714



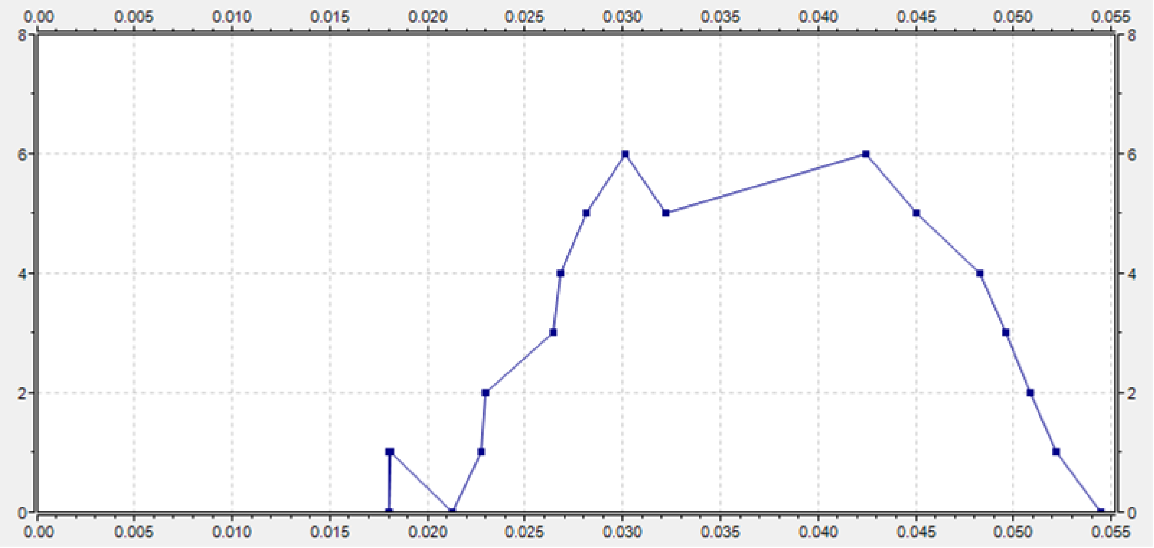
Graph 9: LSR3.ppp1, mean =0.5



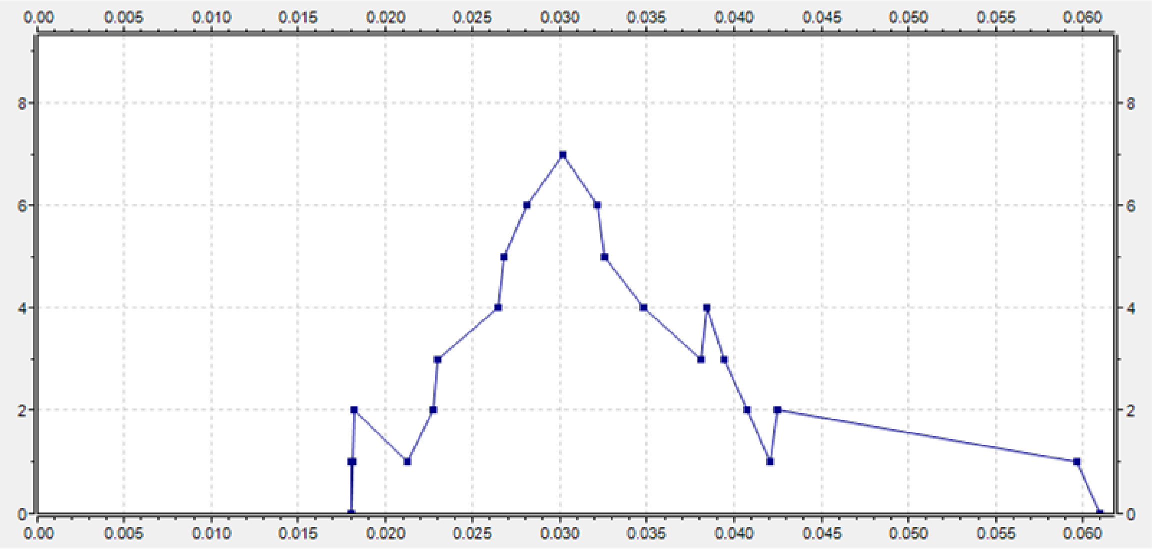
Graph 10: LSR3.ppp2, mean=1.5



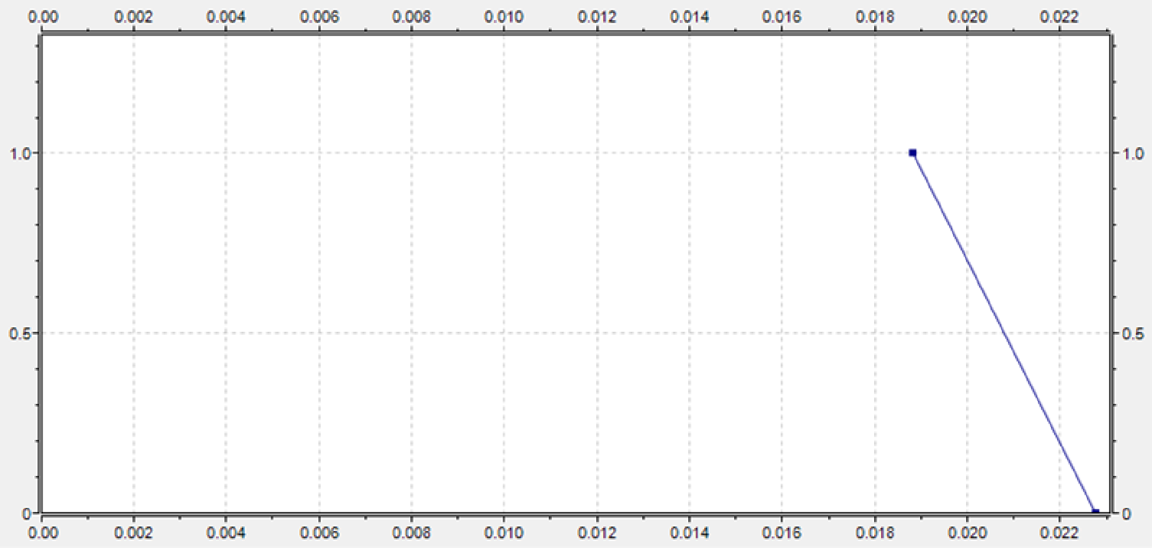
Graph 11: LSR5.ppp3, mean=2.7222



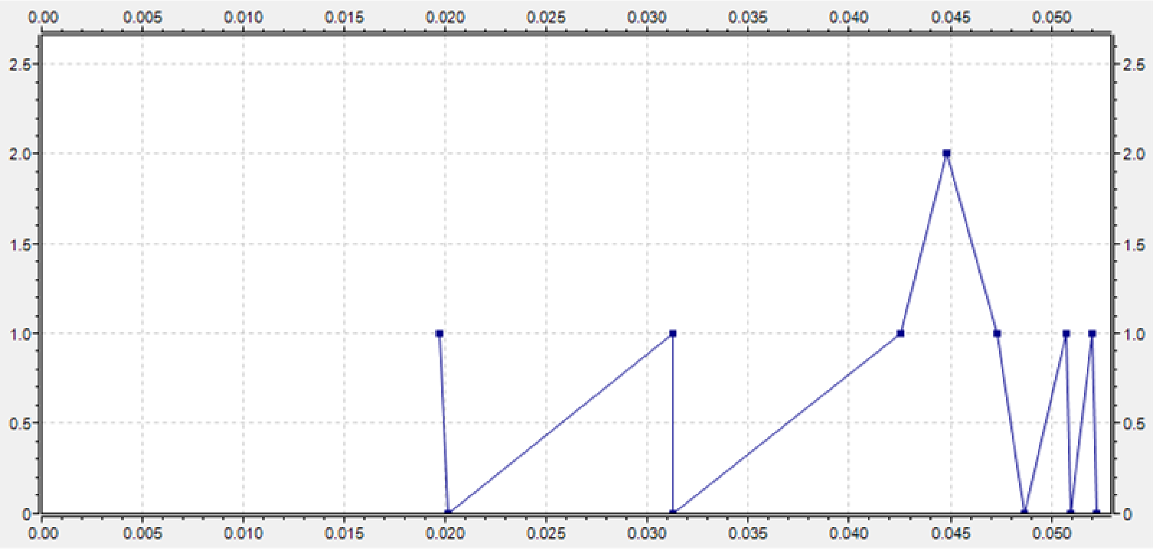
Graph 12: LSR5.ppp4, mean =2.8636



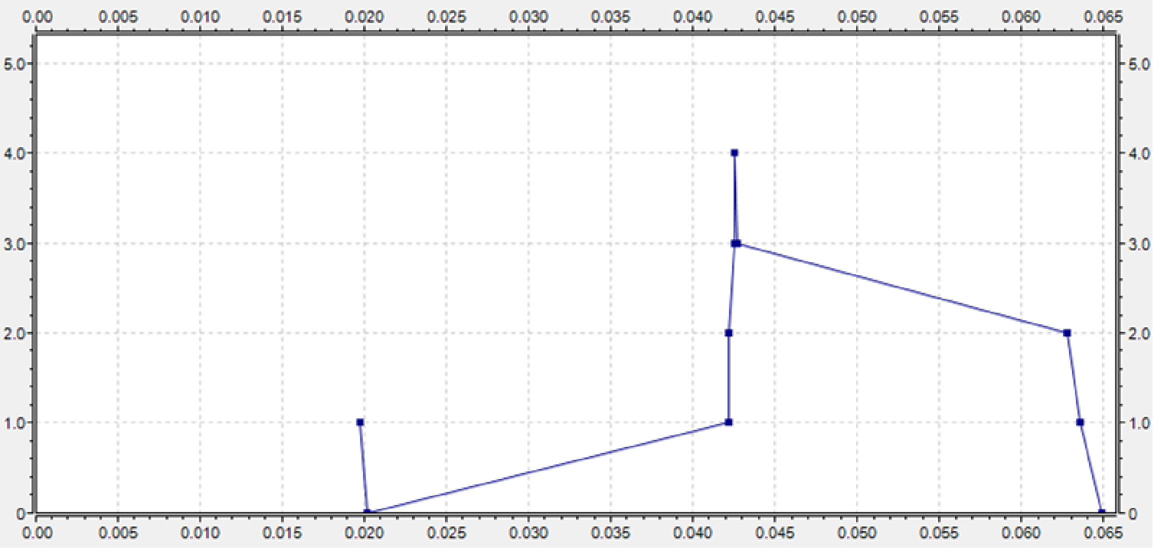
Graph 13: LSR1.ppp1, mean=0.5



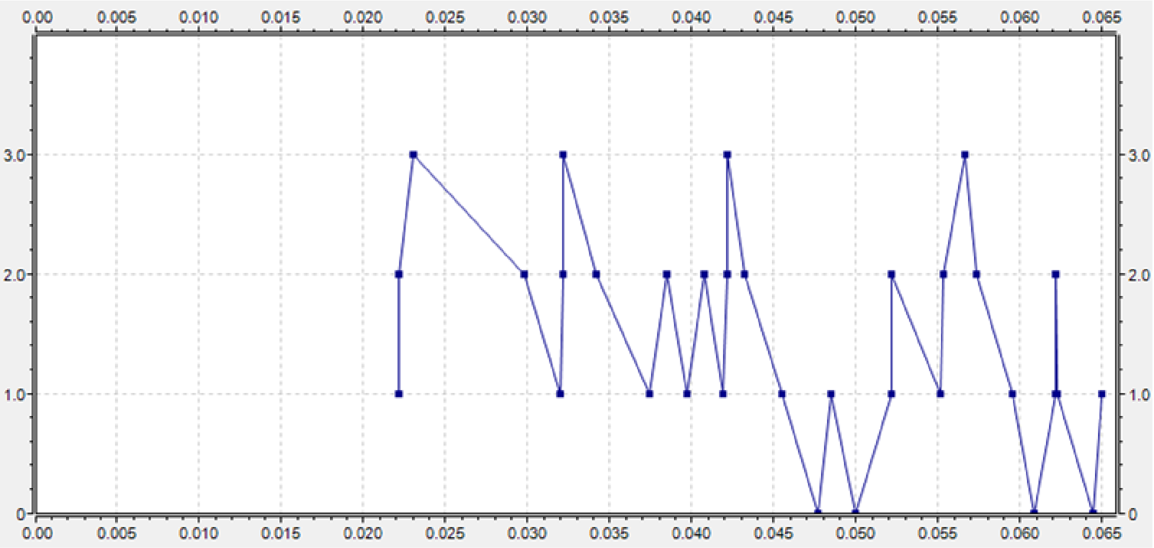
Graph 14: LSR7.ppp0, mean=0.6666



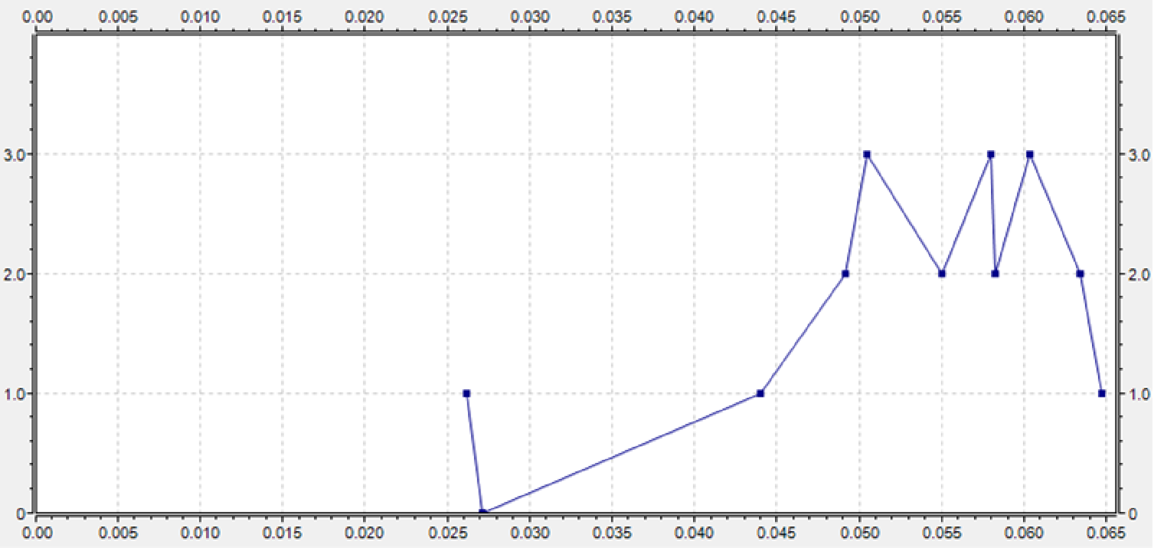
Graph 15: LSR7.ppp1, mean =1.7



Graph 16: LSR1.ppp0, mean =1.4848



Graph 17: LSR6.ppp1, mean =1.8181



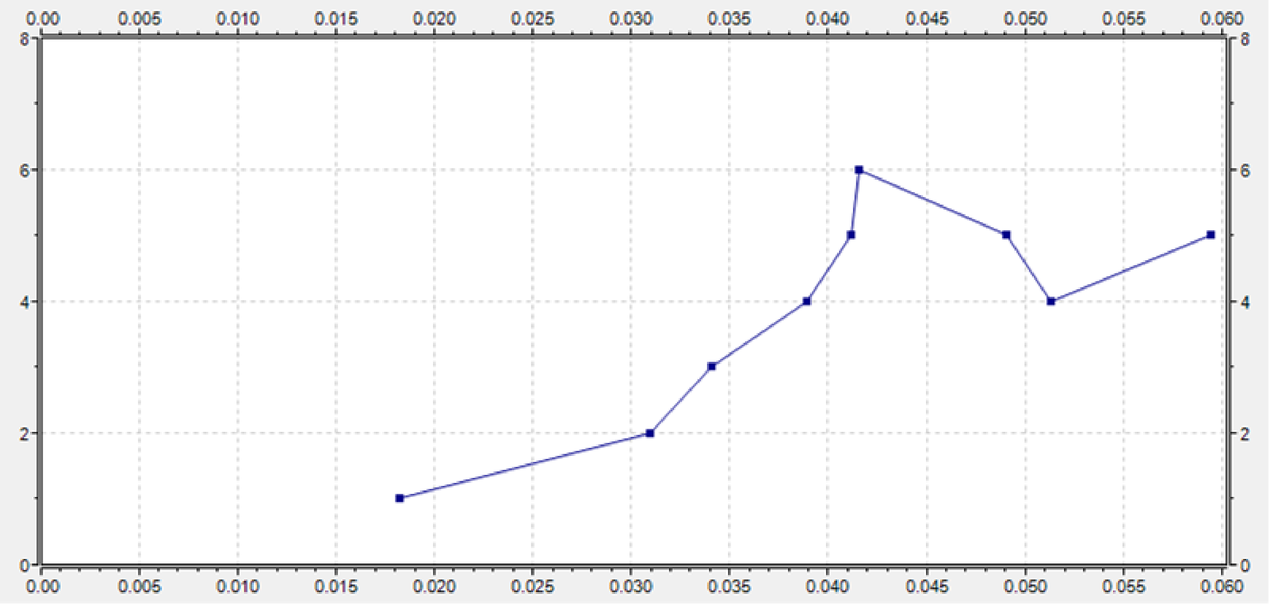
CASE 2:

Queue length for interface ppp4 of LSR5 for different datarates.

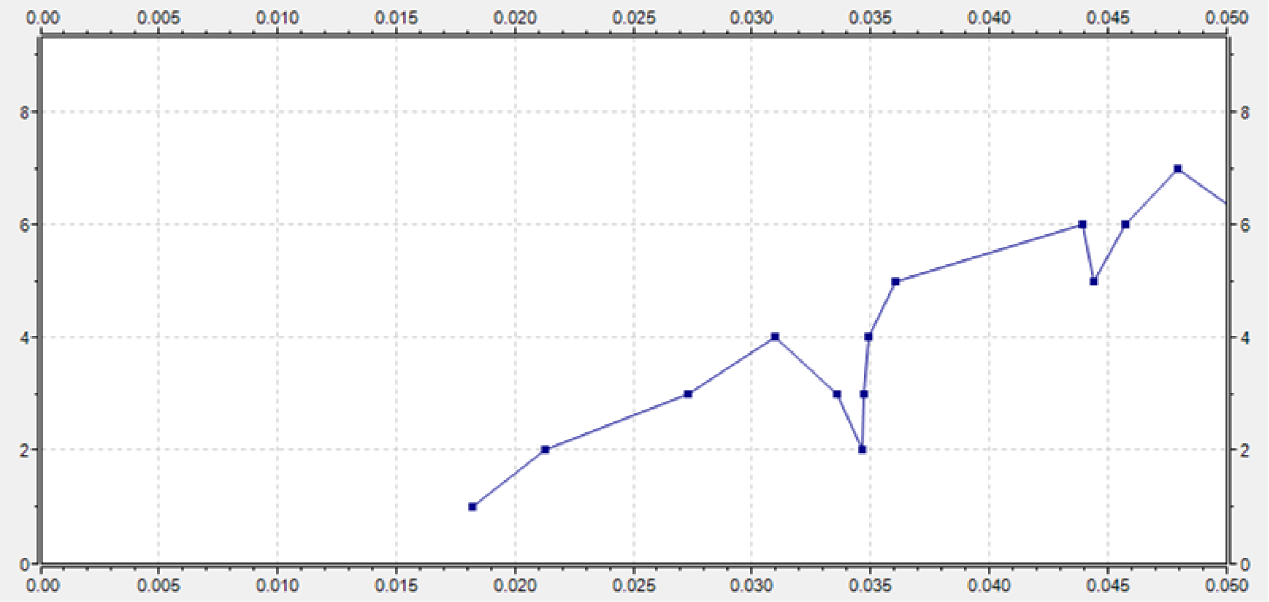
X-axis: time elapsed as an unit defined in the OMNET++ simulation environment (0.001 units = 1 sec)

Y-axis: number of packets being queued at the interface at that time

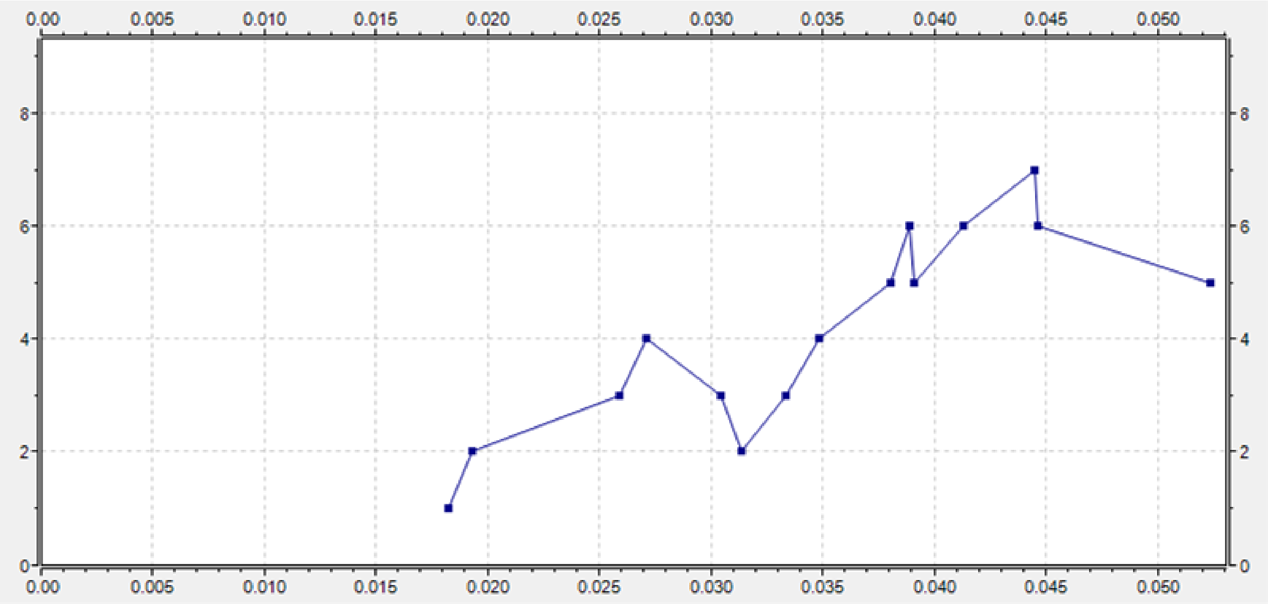
Graph 18 : Datarate = 100kbps; mean=3.8888



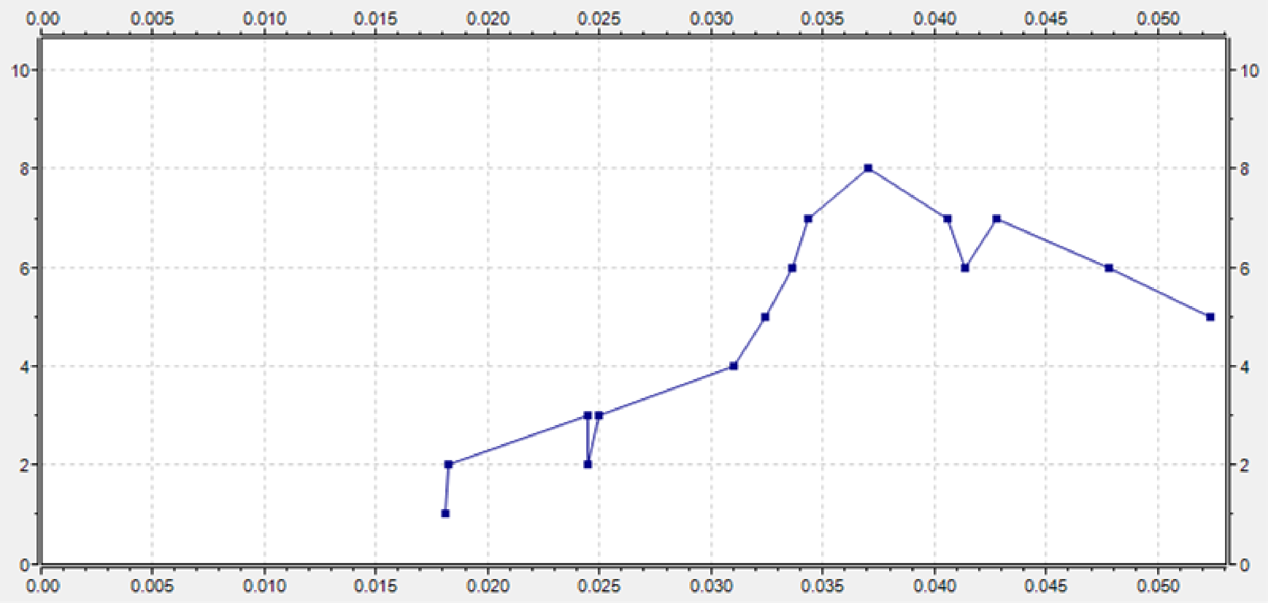
Graph 19 : Datarate=200 kbps, mean=4.0714



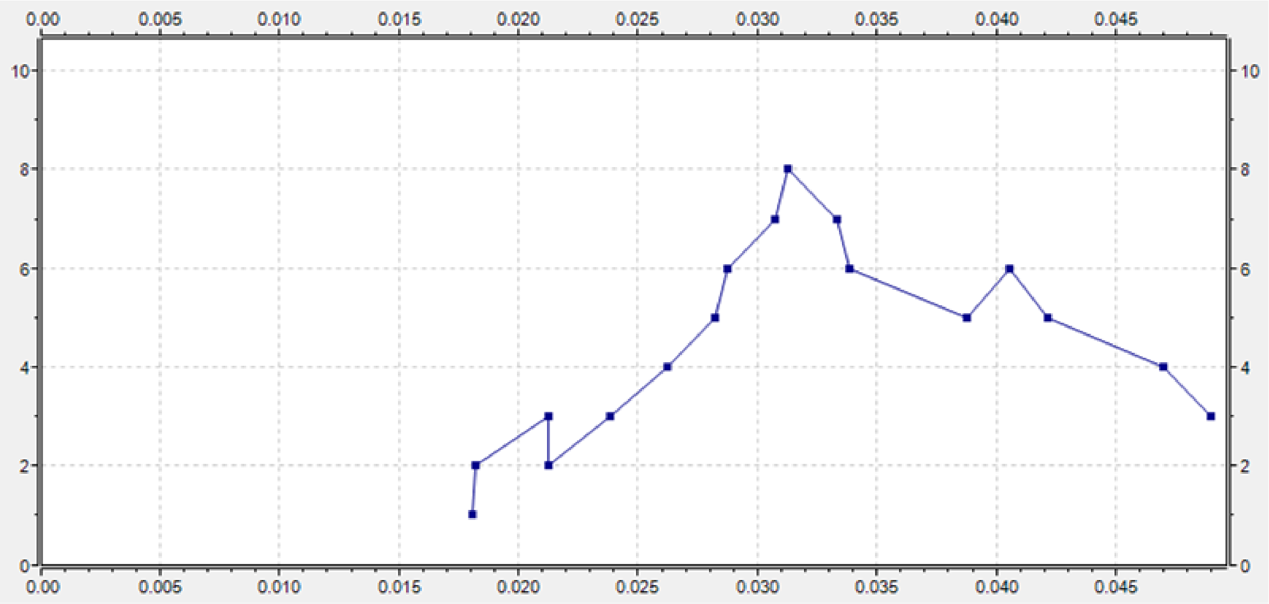
Graph 20 : Datarate = 250 kbps; mean= 4.1333



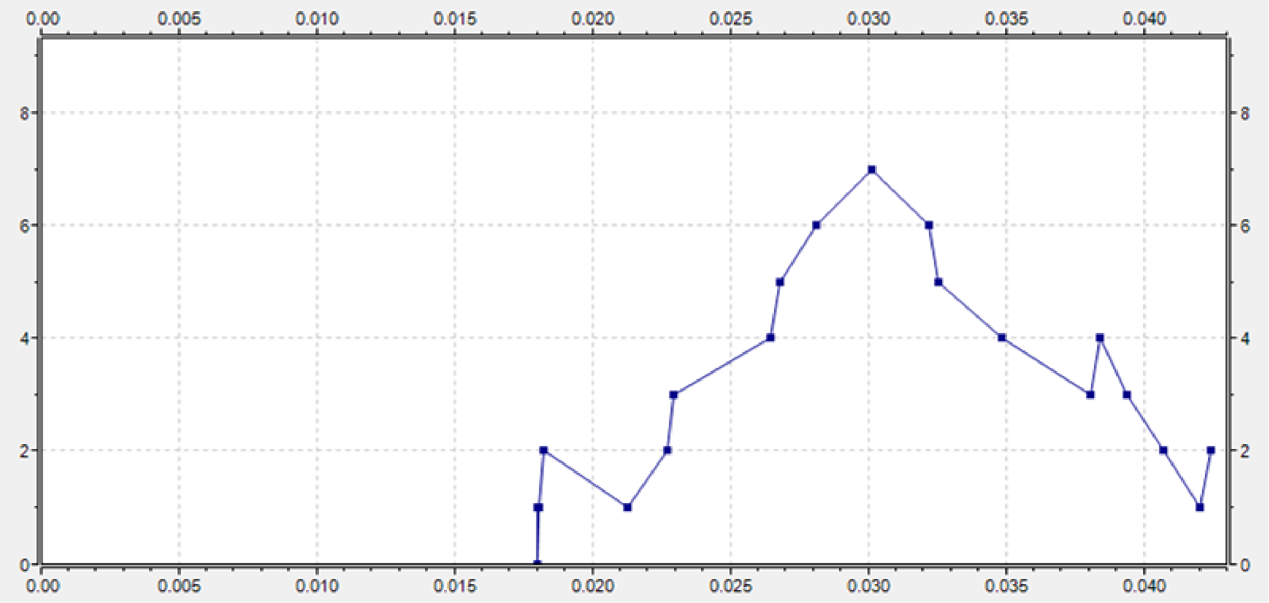
Graph 21 : Datarate = 300kbps; mean=4.8



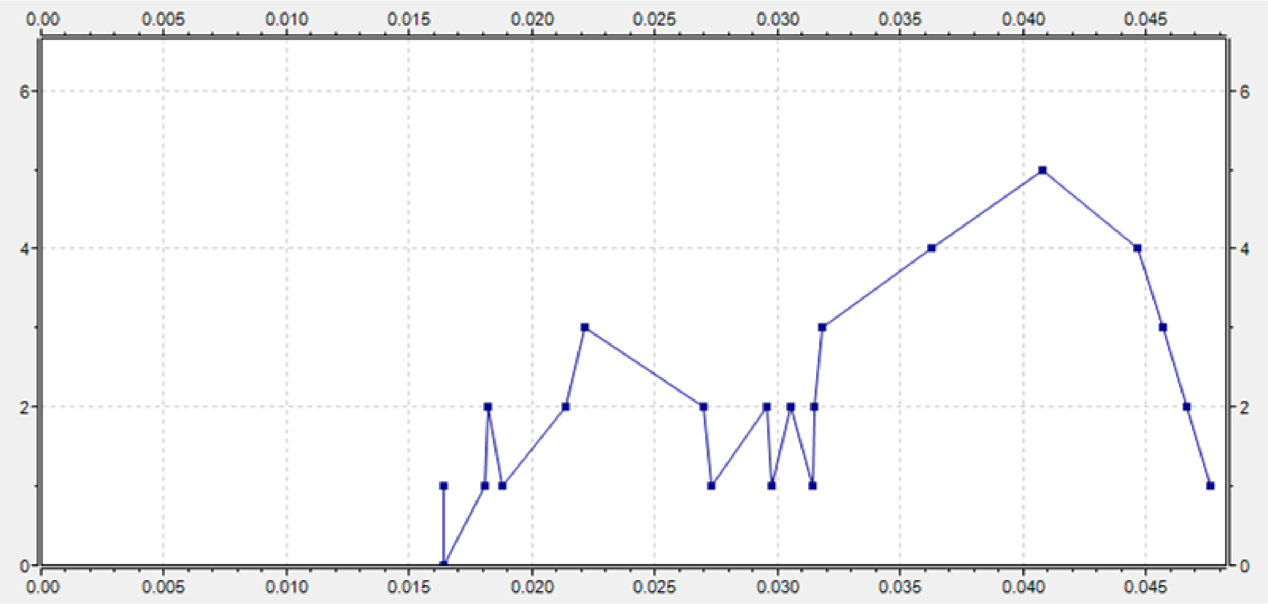
Graph 22 : Datarate = 400 kbps; mean = 4.5294



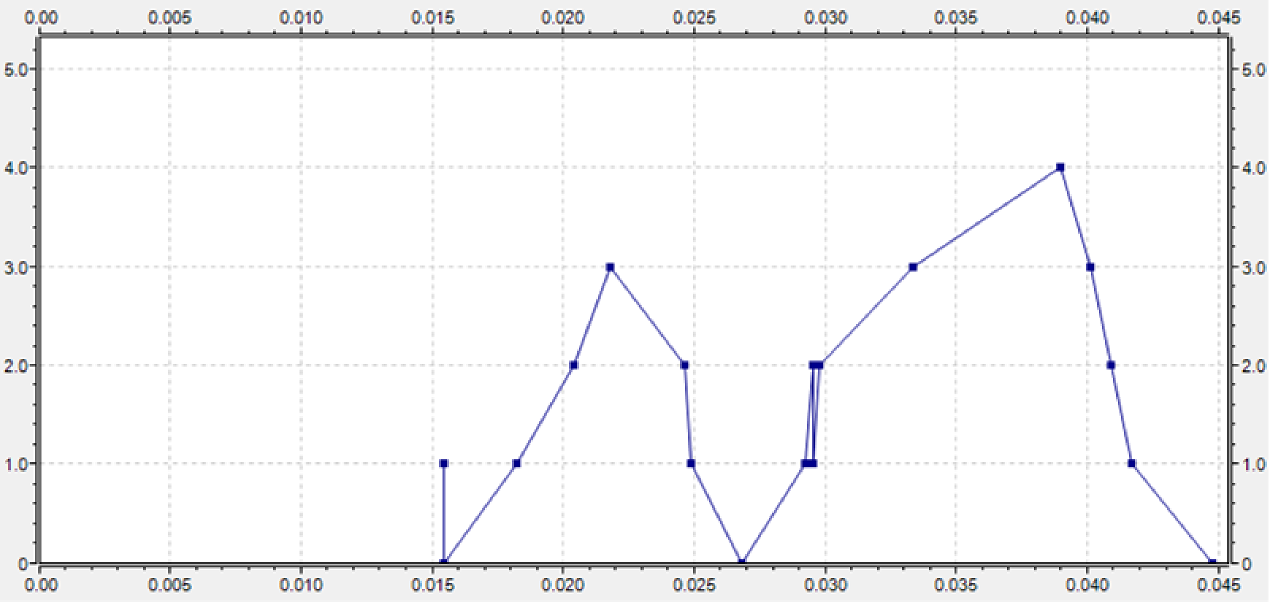
Graph 23 : Datarate = 600kbps;mean=3.1



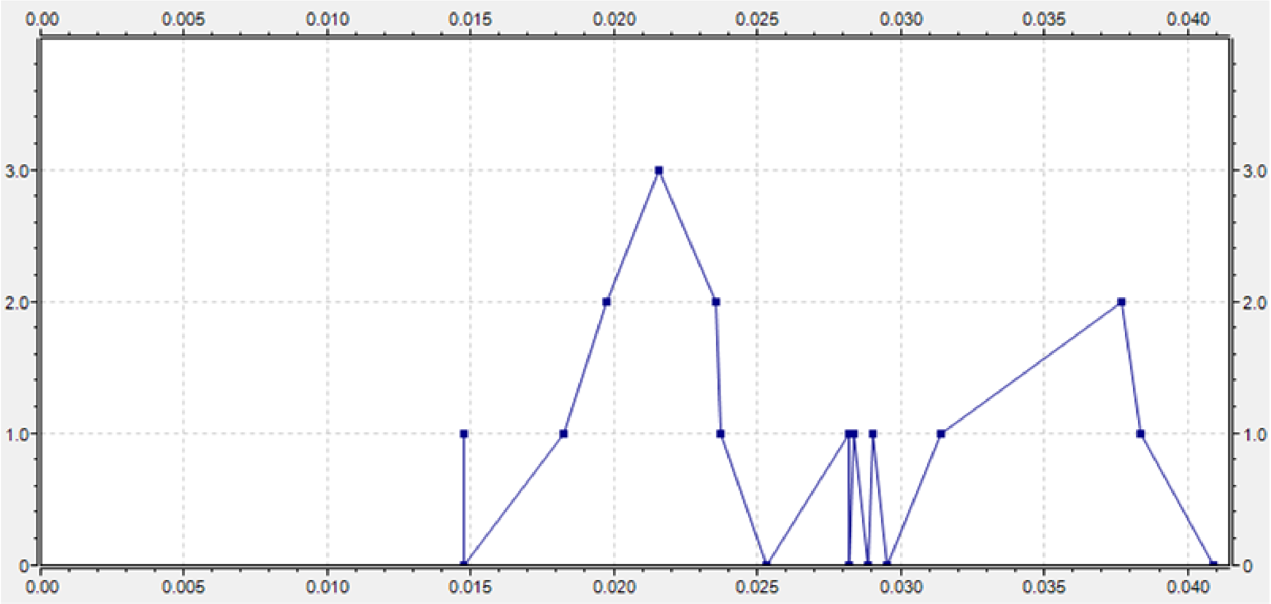
Graph 24 : Datarate = 800 kbps;mean=2.0476



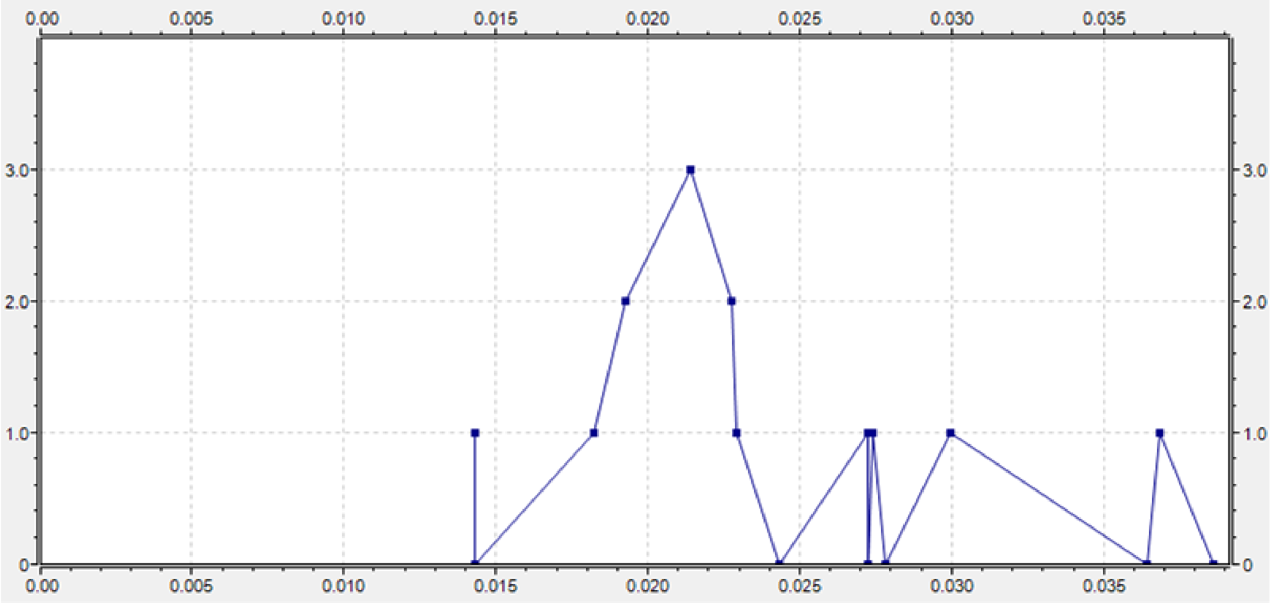
Graph 25 : Datarate = 1000 kbps; mean = 1.6111



Graph 26 : Datarate =1200kbps; mean = 0.9444



Graph 27 : Datarate = 1400 kbps; mean = 0.875



Chapter 8

ANALYSIS

Plotting the mean values at different datarates for CASE 2.

X-axis: datarate in kbps.

Y-axis: Mean queue length of LSR5.ppp[4] for that datarate.

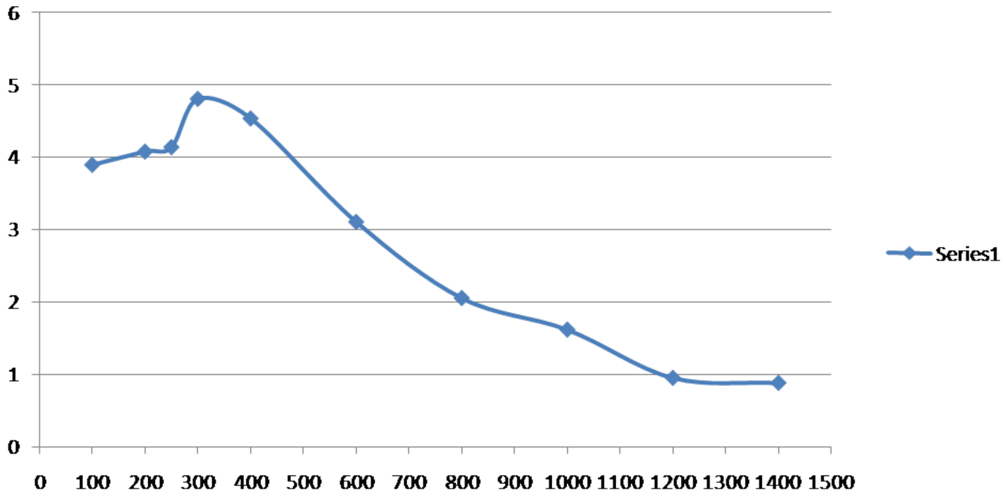


Figure 8.1: *Plot of queue length of ppp4 of LSR5 for different datarate values*

8.1 Observation and Inference:

From this graph, it is seen that as datarate increases, the queue length first increases for small values of datarate and then steadily decreases. The inference we draw from this trend is that, as datarate of a channel increases, the amount of traffic it carries at a time increases and so the traffic that accumulates at the starting node of the channel, decreases. This means that the number of packets being queued at the starting node reduces. This explains the steady decrease in the queue length i.e. the amount of traffic accumulating at a node (here LSR5), as datarate of the channel increases. The above graph gives the trend for interface ppp4 of LSR5. LSR5 is the only node through which every packet needs to pass to reach any of the destination hosts(host3 and host4). It is, thus, a bottleneck link and so we have based our readings on this particular Label Edge Router.

Chapter 9

CONCLUSION

As we can see from the analysis given above, the queue length [1] of a node depends heavily on the datarate of the channel through which it forwards data. A bandwidth allocation strategy where the main concern is congestion avoidance can thus be optimized by considering the instantaneous data rate. The queue length parameter is a direct implication of congestion at a node. So an attempt to monitor the datarate over an interval and make bandwidth reallocation accordingly is our motive. Our algorithm suggests a mechanism that helps in a better optimized usage of available bandwidth in an MPLS network. We try to incorporate our idea to the algorithms proposed in [2] and [3]. By considering the actual data rate over an interval, the bandwidth allocated on the basis of an approximation is made more accurate so as to free some unused bandwidth to make space for new LSPs. This improves the overall utilization of the available bandwidth. The above mentioned improvement is only theoretical.

Moreover, the project addresses the issue of optimized bandwidth allocation problem by taking into account the fluctuations in the bandwidth requirement of LSPs. In this project, thus we have put together the concept of changing bandwidth reuirement of LSPs and changing data rate to improve the usage of bandwidth in the network.

REFERENCES:

[1]Christopher Marty, Mohamed A.Ali, "A System for Weighted Max-Min Congestion Control in MPLS Networks", Journal of Network and Systems Management, VOL.14, NO.4, DECEMBER 2006, , pg. 537-542.

[2]C.-L. Lee, C.-W. Chen, and Y.-C. Chen, "Weighted proportional fair rate allocations in a differentiated services network",IEICE Trans. Commun., VOL.E85-B, NO.1, pg. 115-128, JANUARY 2002.

[3]Teruaki Yokoyama, Katsuyoshi Iida, Hiroyuki Koga, Suguru Yamaguchi, "Proposal for Adaptive Bandwidth Allocation Using One-Way Feedback Control for MPLS Networks", IEICE Trans. Commun., VOL.E90-B, NO.12, DECEMBER 2007, pg. 3530-3534.

[4]Xipeng Xiao, Alan Hannan, Brook Bailey, Lionel M. Ni, "Traffic Engineering with MPLS in the Internet", IEEE Network Magazine, VOL.14, 2000, pg. 2-4.

[5]Ayan Banerjee, John Drake, Jonathan Lang, Brad Turner, Daniel Awduche, Lou Berger, Kireeti Kompella Yakov Rekhter, "Generalized Multiprotocol Label Switching: An Overview of Signaling Enhancements and Recovery Techniques", IEEE Communications Magazine, JULY 2001, pg. 144-148.

[6]Pramoda Nallur, "Multi-Protocol Label Switching (MPLS)", Alcatel Internetworking Division, TECON 2000, 29 JULY 2000.

[7]Chuck Semeria, "Multiprotocol Label Switching : Enhancing Routing in the NewPublic Network", Juniper Networks, Inc., Part Number :200001-002, 27 SEPTEMBER 1999, pg. 13.

[8]http://www.juniper.net/techpubs/en_US/junos10.0/information-products/topic-collections/nog-mpls-model/mpls-lsp-state-determining.html

[9]Manual, INET Framework for OMNET++

[10]J. Wang, S. Patek, H. Wang, and J. Liebeherr, “Traffic Engineering with AIMD in MPLS networks”, Protocols for High-Speed Networks, 2002, pg. 1-4

[11]“MPLS”, http:// www.wikipedia.org