Steady-State Performance Analysis of MPLS Label Switching[†]

Ling-Chih Kao
Graduate Institute of
Communication Engineering
National Taiwan University
TEL: 886-2-23635251 Ext. 554
FAX: 886-2-23683824
Email: d6942004@ms.cc.ntu.edu.tw

Zsehong Tsai
Department of Electrical Engineering

National Taiwan University
TEL: 886-2-23635251 Ext. 543
FAX: 886-2-23683824
Email: ztsai@cc.ee.ntu.edu.tw

Abstract

In this paper we propose a close-loop queueing model of MPLS switch with the different label-setup and release policies, supporting both traffic-driven and topology-driven connection setup procedures. This model can emulate the behavior of TCP under the MPLS switch when the maximum window size is sustained and the packet loss rate is negligible. From the proposed flow-based MPLS switch model, one can clearly observe the competition of multiple IP flow for limited labels, and how the label-setup policy and the label-release policy affect the system performance. We find that Norton's theorem can be applied to solve this sophisticated queueing model. Therefore, with very limited computational complexity with respect to the number of IP flows or labels, the proposed mathematical model and the approximation of label competition can be used to obtain the desired performance metrics, such as the throughput, the label-setup rate, and the channel utilization. Finally, the trade-off among performance metrics can be observed as well.

key words: flow, MPLS, performance analysis.

1 Introduction

With the advent of fiber optics technologies, the network bottleneck has shifted from the shortage of bandwidth to the processing capability of the router. The label switch is found often superior to the traditional IP router since not all packets belonging to a flow require the processing of IP routing engines. After some packets are identified as leading packets of a flow, then all their following packets can be switched directly by a layer-2 switching technique, such as label-based switching. In other words, new techniques in label switching take advantage of IP routing and high speed switching and can save a lot of packet transmission time over the network. The Internet Engineering Task Force (IETF) thus adopted the MultiProtocol Label Switching (MPLS) architecture [1, 2] which can be used for transport of any level 3 protocol over any level 2 technology. The basic operation of MPLS is that only the packet passes through the edge LSR (Label-Switching Router) needs performing a conventional longest-match lookup in the IP header, the forwarding mechanism that the packet passes through the core LSR is based only on label swapping. In general, MPLS switches have more powerful traffic engineering capability [3, 4] than the traditional router. However, the resources (labels) in the MPLS switch are often limited. Some flow aggregation methods [5] are then proposed to save the label usage and have triggered research work on related engineering or modeling.

In label or IP switch models proposed in previous research works, only a few of them has been used to evaluate the trade-off between the label-setup policy and the channel (link) utilization or throughput performance. The label-setup policy proposed in [6] is that if the accumulated packets in the switch buffer have exceeded the triggering threshold within a fixed time interval, the IP controller shall set up a label. The label-setup policy proposed in [16] is that either if the accumulated packets in the switch buffer have exceeded the triggering threshold or if the label-setup timer has expired, the IP controller shall set up a label. The label-setup policies in [6, 16] are approximately similar under heavy traffic loads. The label-release policy is discussed more frequently (see [9, 10, 16]). The label can be released only during the maximum allowed inactive duration, if no packets arrive. Our work [16] presented an efficient model for investigating the related performance issues. However, it is restricted to the single IP flow. Although the mathematical models proposed in [11, 12] allowed multiple flow, they did not describe the detailed operation of the label-setup and release policies.

From [8], we know that most Internet traffic processed by IP switches is long-lived, which should also hold for MPLS switches. So one can focus on the effects of long-duration traffic passing through an MPLS switch. We propose to use the closed-loop queueing network model with the round-trip delay, assuming the maximum TCP window size is maintained during the communication. In this paper, we extend our previous work [16] to model the behavior of multiple IP flow that contend for a limited set of labels. The performance of the traffic-driven and the topology-driven MPLS switch with various label-setup and release policies can then be evaluated by such queueing model. Since it is difficult to directly manipulate an original but sophisticated queueing model, we introduce the concept of label vacation model to approximate the availability of label in a label-competition scenario. With the approximation approach for a multiple IP flow MPLS model, one can obtain the results quickly without resorting to simulation. With this model, one can also investigate the effects of the label-setup timer and the label-release timer on a flow-based MPLS model, and observe the trade-off among the throughput, the label-setup rate, and the channel utilization. Furthermore, our proposed queueing model could be applied to the latest proposed "λ-labeling" MPLS system [7] if the wavelength-switched technique become available.

The remainder of the paper is organized in the following. In Section 2, the queueing model for a flow-based MPLS switch is described. In Section 3, an approxi-

[†] This work was supported by National Science Council of the Republic of China under Grant NSC 89-2213-E-002-168 and by Ministry of Education under Grant 89E-FA06-2-4-7.

0-7695-0951-7/01 $10.00 © 2001 IEEE

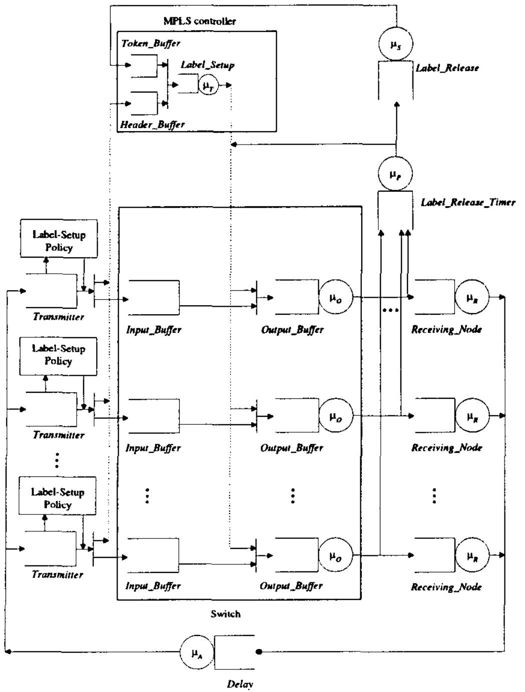

Figure 1: Original queueing model.

mate and efficient analysis is presented. In Section 4, four performance measures are derived. Since estimating an adequate label vacation time becomes the key factor that determines the success of the proposed model, the method of estimating an adequate vacation time is proposed in Section 5. Numerical examples and simulation results are presented in Section 6. Conclusions are drawn in Section 7.

2 Queueing Model

In this section, we present a queueing model which characterizes the behavior of N IP flows competing for M labels in a flow-based MPLS switch. Since the objective is to model the behavior of flow-based MPLS under heavy traffic loads, the employed model is close-loop queueing network model. A fixed window size for each flow is assumed.

The detailed queueing network, called the original model, is shown in Figure 1. Ten queueing nodes are included in this model: *Token_Buffer*, *Header_Buffer*, *Label_Setup*, *Label_Release*, *Label_Release_Timer*, *Transmitter*, *Input_Buffer*, *Output_Buffer*, *Receiving_Node* and *Delay*. The *Transmitter*, the *Input_Buffer*, the *Output_Buffer*, the *Receiving_Node* and the *Delay* queue form a loop for the IP data packet and IP ack. The *Token_Buffer* stores the tokens which represent the availability of the channel. The channel is available if there is a token in the *Token_Buffer*. The *Header_Buffer* stores the header information of the first packet of each IP flow when it requests for a label. The *Label_Setup* queue represents the time required to look up the routing/switching table and set up a label. The *Label_Release* queue represents the time required to release a label. The *Label_Release_Timer* queue represents an label-release timer. This timer indicates the maximum length of idle period of a flow before its label is released for other use. The *Delay* queue represents the total round-trip delay of the forward and the feedback channels. Once a flow is granted a label, the flow is served with its own output server and uses its own label-setup and release mechanism. The switch output is assumed to have enough buffers, so the packet loss rate is negligible. The detailed operations are described as follows.

When a burst of packets for a flow arrive, the header information of the first packet for that flow is extracted and copied into the *Header_Buffer* of the MPLS controller. The leading packet and its following packets then enter into the *Input_Buffer* of the switch. If the condition that a token resides in the *Token_Buffer*, and the accumulated packets reside in the *Input_Buffer* of the switch has exceeded the triggering threshold or the label-setup timer has expired, the MPLS controller then starts to set up a label. The switch starts serving packets when the label has been set up. The label manager maintains an activity timer to control the label-release operation. The label is released only if the activity timer indicated that the maximum allowed inactive duration has been reached.

The original queueing model is too complicated to solve directly. The behavior of the original queueing model of each flow can be observed to be of two kinds. The first kind ($N \leq M$) is that a flow can get a label immediately when it requests for a label. Concerning this part, one can refer to [16] for detail. This happens when labels are available to all flows. The second kind ($N > M$) is that a flow can not be allocated a label immediately when it requests for a label. The analysis of this part is described in the next section. This happens when labels are not enough for all flows. Therefore, one can consider a label is on vacation when a flow requests for a label, but no label is available. In other words, the impact of label competition on the behavior of a single IP flow can be modeled via a label *vacation* node. As a result, the steady-state queueing model in Figure 2 is proposed for performance analysis.

In this model, we still use Erlang-k distribution in *Label_Release_Timer* to approximate a constant label-release timer. Erlang-k distribution is employed to approximate the label vacation time. An infinite server queue is used for the *Delay* queue to approximate the round-trip delay. In all nodes, service time is assumed to be exponentially distributed, unless stated otherwise.

3 Steady-State Analysis

We here present a procedure to efficiently calculate steady-state distribution of the proposed flow-based MPLS switch model as shown in Figure 2, with a fixed window size W for each IP flow. We adopt the following notations:

W : maximum window size, in packets.

μ_A : service rate in the *Delay* queue.

μ_O : service rate in the *Output* queue.

μ_R : service rate in the *Receiving_Node* queue.

$T_{rel} = \frac{1}{\mu_P}$: the average sojourn time in the *Label_Release_Timer* queue.

k : the number of phases of the Erlang-k distributed sojourn time in the *Label_Release_Timer* queue.

T_{set} : the threshold of the label-setup timer.

μ_S : service rate in the *Label_Release* queue.

μ_T : service rate in the *Label_Setup* queue.

μ_V : service rate in the label *Vacation* queue.

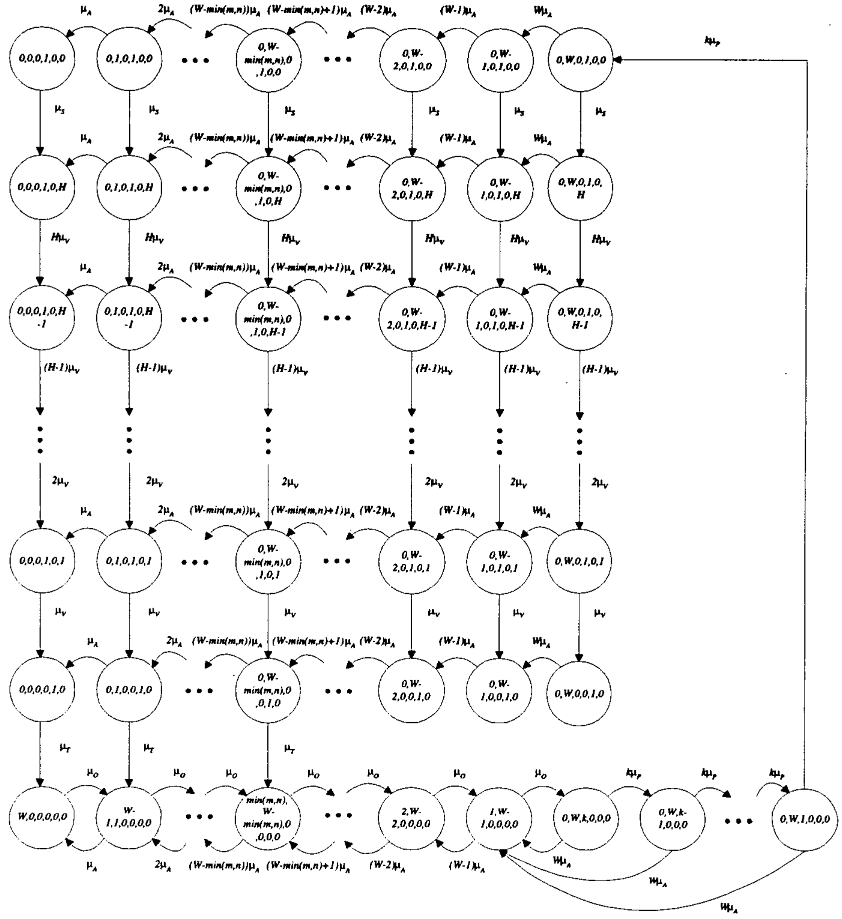

Figure 3: State-transition diagram of the shorted-circuit queueing model.

n_A : the number of packets in the *Delay* queue, ranging from 0 to *W*.

n_O : the number of packets in the *Output_Buffer* queue, ranging from 0 to *W*.

n_S : the number of packets in the *Input_Buffer* of the switch, ranging from 0 to *W*.

n_T : the number of tokens in the *Token_Buffer*, equals 0 if the channel is available, equals 1 if the channel is in use.

π_T : the current phase at which the token stays in the *Label_Release_Timer* queue, ranging from 1 to *k*.

π_R : the state of the *Label_Release* queue, equals 0 if the *Label_Release* queue is idle, equals 1 if label-release has been started.

π_V : the current phase at which the token stays in the *Vacation* queue, ranging from 1 to *H*.

m : the triggering threshold which represents the number of accumulated packets waiting for setting up a label, ranging from 1 to *W*.

Unfortunely, the steady-state queueing model shown in Figure 2 is still overly complicated for deriving exactly solutions. Two steps are required in the approximate performance analysis. First, we cut down the model shown in Figure 2 so that the equivalent Norton's theorem in queueing analysis (see [13, pp. 250] or [14]) can be employed to carry out the approximate calculation. The procedure of this simplification is in the following: we short points E and F of the model in Figure 2 to obtain the shorted-circuit queueing model of the steady-state queueing model. We then employ the state vector (a, b, c, d, e, f) to represent the state of the shorted-circuit model when $n_O = a, n_A = b, \pi_T = c, \pi_R = d, n_T = e$, and $\pi_V = f$.

Based on the above definitions, the global balance equations of shorted-circuit queueing model are listed as follows.

$$H\mu_V P_{0,W,0,1,0,1} = W\mu_A P_{0,W,0,0,1,0} \tag{1}$$

$$\mu_S P_{0,W,0,1,0,0} = (W\mu_A + H\mu_V)P_{0,W,0,1,0,H} \tag{2}$$

$$H\mu_V P_{0,W,0,1,0,i+1} = (W\mu_A + H\mu_V)P_{0,W,0,1,0,i} \;, \quad 1 \le i \le H-1 \tag{3}$$

$$k\mu_P P_{0,W,1,0,0,0} = (\mu_S + W\mu_A)P_{0,W,0,1,0,0} \tag{4}$$

$$k\mu_P P_{0,W,i+1,0,0,0} = (k\mu_P + W\mu_A)P_{0,W,i,0,0,0} \;, \quad 1 \le i \le k-1 \tag{5}$$

$$\mu_O P_{1,W-1,0,0,0,0} = (k\mu_P + W\mu_A)P_{0,W,k,0,0,0} \tag{6}$$

$$\mu_S P_{0,i,0,1,0,0} + (i+1)\mu_A P_{0,i+1,0,1,0,H} = (i\mu_A + H\mu_V)P_{0,i,0,1,0,H} \;, 0 \le i \le W-1 \tag{7}$$

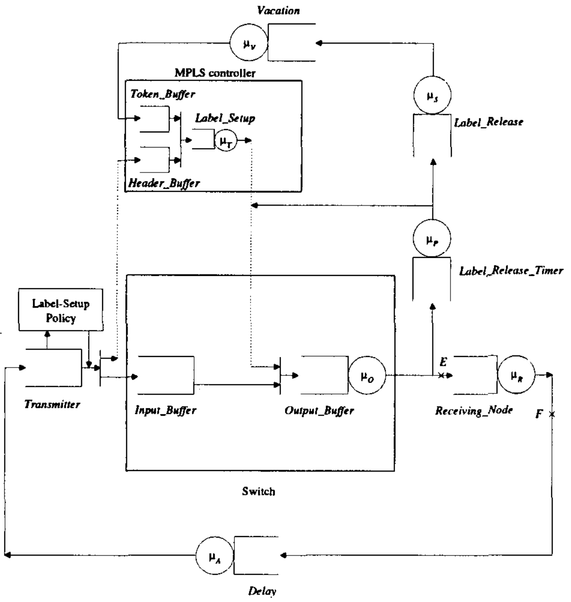

Figure 2: Steady-state queueing model.

$$
\begin{aligned}
&H\mu_V P_{0,i,0,1,0,j+1} + (i+1)\mu_A P_{0,i+1,0,1,0,j} = \\
&(i\mu_A + H\mu_V)P_{0,i,0,1,0,j} \; ; 0 \le i \le W-1 \;, \\
&1 \le j \le H-1 && (8) \\
&(i+1)\mu_A P_{0,i+1,0,1,0,0} = (\mu_S + i\mu_A)P_{0,i,0,1,0,0} \; ; \\
&0 \le i \le W-1 && (9) \\
&(i+1)\mu_A P_{0,i+1,0,0,1,0} + H\mu_V P_{0,i,0,1,0,1} = \\
&(i\mu_A + \mu_T)P_{0,i,0,0,1,0} \;, \\
&0 \le i \le W - min(m,n) && (10) \\
&(i+1)\mu_A P_{0,i+1,0,0,1,0} + H\mu_V P_{0,i,0,1,0,1} = \\
&i\mu_A P_{0,i,0,0,1,0} \; ; \\
&W - min(m,n) + 1 \le i \le W-1 && (11) \\
&(\mu_O + j\mu_A)P_{W-j,j,0,0,0,0} = \mu_T P_{0,j,0,0,1,0} + \\
&\mu_O P_{W-j+1,j-1,0,0,0,0} + (j+1)\mu_A \cdot \\
&P_{W-j-1,j+1,0,0,0,0} \; ; 1 \le j \le W - min(m,n) && (12) \\
&(\mu_O + j\mu_A)P_{W-j,j,0,0,0,0} = \mu_O P_{W-j+1,j-1,0,0,0,0} \\
&+(j+1)\mu_A P_{W-j-1,j+1,0,0,0,0} \; ; \\
&W - min(m,n) + 1 \le j \le W-2 && (13) \\
&\mu_O P_{W,0,0,0,0,0} = \mu_T P_{0,0,0,0,1,0} + \mu_A \cdot \\
&P_{W-1,1,0,0,0,0} && (14)
\end{aligned}
$$

where $P_{a,b,c,d,e,f}$ is the steady-state probability of the state vector (a, b, c, d, e, f). In equations (10)–(13), n is the integer such that $\frac{1}{W\mu_A} + \frac{1}{(W-1)\mu_A} + \ldots + \frac{1}{(W-n+1)\mu_A}$ is closest to T_{set}. This is due to a released flow must have all its packets in the backlog at the *Delay* queue in the short-circuited model. In other words, instead of directly modeling T_{set}, we use a corresponding pseudo threshold n to limit the holding time of packets. The state-transition diagram of the shorted-circuit queueing model corresponding to equations (1)–(14) is shown in Figure 3. In most cases, we set $n > m$. Consequently, in the balance equations, the label-setup timer is dominated by a triggering threshold m so that a label is set up mostly when the number of waiting packets have reached m and the timer has almost no effect (i.e. for protection only). Since the balance equations are used to obtain steady-state performance, therefore, this approximation should hold well. After examining this state-transition diagram, one can conclude from Kolmogorov's criterion that the underlying system process is not time reversible and it does not satisfy the quasi-reversibility [13]. Hence, product-form solution can not be easily obtained since local balance equations do not hold. Fortunately, the structure of this diagram follows certain rule. We found an iterative computational algorithm which can solve global balance equations effectively even for large state space.

Let $\eta(w)$ denote the throughput of the shorted-circuit queueing model, provided that the population (window size) is w. Then $\eta(w)$ can be calculated using the steady-state solution obtained via the above calculation procedure and the following equation:

$$\eta(w) = \mu_O \sum_{i=1}^{w} P_{i,w-i,0,0,0,0} \;, w = 1, 2, \ldots, W \tag{15}$$

The throughput $\eta(w)$ can be used for approximating the performance of the steady-state model as follows. Applying the procedure as used in the equivalent Norton's theorem, the shorted-circuit queueing model is then replaced by an *equivalent* queue. Therefore, the steady-state queueing network can be reduced to an approximate queueing model.

The equivalent queueing network is now a typical product-form queueing network. The second step of our analysis is simply to solve this product-form queueing network, which can be solved via well-known techniques available in [15].

4 Performance Measures

The steady-state throughput, the label-setup rate and the channel utilization are considered to be key performance metrics. The steady-state throughput is reflected by the average cycle time in the steady-state queueing model. The label-setup rate is concerned with the required label processing overhead. With the channel utilization, one can predict the ratio of wasted bandwidth. The formula of these metrics are as follows.

4.1 Average Cycle Time and Throughput

The average cycle time T_C, the time required by a packet or its ack to circulate around the message loop in the steady-state queueing model, is proportional to the inverse of throughput. It is well-known that

$$T_C = \frac{WP_{size}}{\eta^*(W)} \tag{16}$$

where P_{size} is the packet size, and $\eta^*(W)$ is the throughput of equivalent queueing networks given the maximum window size W.

4.2 Label-Setup Rate

The label-setup rate S_R is defined as the average number of label-setup operations in the MPLS controller per unit time. Using the steady-state probability that the label-setup token is ready, one can derive the label-setup rate S_R as

$$S_R \approx \mu_T \sum_{i=0}^{W-min(m,n)} P_{0,i,0,0,1,0} \tag{17}$$

Here, we assume the capacity of the *Receiving_Node* is much larger than $\eta(W)$. If μ_R is not much larger than $\eta(W)$, equation (20) still serves as a worse case (upper bound) approximation.

4.3 Channel Utilization

The channel utilization U_R is defined as the ratio of the time during the period that the switch serves packets to the time during the period that the channel is reserved. It is given by

$$\begin{aligned} U_R \quad \approx \quad & \sum_{i=1}^{W} P_{i,W-i,0,0,0}/(\sum_{i=1}^{W} P_{i,Wi,0,0,0} + \sum_{i=1}^{k} P_{0,W,i,0,0} + \\ & \sum_{i=0}^{W} P_{0,i,0,1,0}) \end{aligned} \tag{18}$$

The time periods considered to be "reserved" by a flow include the packet transmission time, the idle period waiting for label-release timeout and the time required to release a label. But only during the period that the packet transmission in the channel is considered to be efficiently utilized. Here, we assume the capacity of the *Receiving_Node* is much larger than $\eta(W)$. If μ_R is not much larger than $\eta(W)$, equation (21) serves as a best case approximation.

5 How to determine vacation time

Estimating an adequate vacation time is the key step that determines the accuracy of the proposed model. We have a three-step approach to determine the vacation time. First, we use the token cycle time (equals $1/S_R$) obtained from our previous work [16] or from the model of this paper (via equation (19)) with the vacation time has to be set to be negligible. Before we obtain an accurate estimation, we first assume that if a token is not available, one must wait for an integer multiple of token cycle time. When the number of labels is less than the number of IP flows, some flows can get a label for its own use immediately, while others probably wait 1 or more cycles of token rotation. Therefore, we can then approximate the lower bound of waiting time, denoted by $T_{w,lowerbound}$, and the upper bound of waiting time, denoted by $T_{w,upperbound}$, under this assumption by the following equations

$$\begin{aligned} T_{w,lowerbound} \quad = \quad & \sum_{i=1}^{\lfloor \frac{N}{M} \rfloor - 1} iM/NS_R + (N - M\lfloor \frac{N}{M} \rfloor) \cdot \\ & (\lceil \frac{N}{M} \rceil - 1)/NS_R \end{aligned} \tag{19}$$

$$T_{w,upperbound} = (\lceil \frac{N}{M} \rceil - 1)/S_R \tag{20}$$

where M is the number of labels and N is the number of flows. Secondly, one can study the extreme case when T_{rel} is exceedingly long. In this case, the number of labels equals the number of IP flows that can use the labels forever no matter how many other flows need to access the labels. The steady-state throughput for this case is equal to $(L/F)\eta^\star(W)_{|L/F=1}$, where L/F is the ratio of the number of labels to IP flows and $\eta^\star(W)_{|L/F=1}$ is the steady-state throughput given that L/F equals 1 (which equals $\eta^\star(W)$ in [16] or equation (17) in this paper if the vacation time is set to be negligible). Hence, one can know the extreme case steady-state throughput for various L/F. Finally, one can select an estimate value of $T_w \in \{T_{w,lowerbound}, T_{w,upperbound}\}$ which satisfies $|\eta^\star(W)_{|T_w} - (L/F)\eta^\star(W)_{|L/F=1}| < \epsilon$, where $T_w = \mu_V^{-1}$ is equivalent to the waiting time for the next available label, $\eta^\star(W)_{|T_w}$ is the steady-state throughput obtained by substituting all the equations in section 3 with μ_V for T_w^{-1}, and ϵ is a small value. The last criterion is based on an approximate conservation law, $S_R T_w = C$ in our system, where C is some constant. For different thresholds value of T_{rel} under the same given L/F, the waiting time for the next available label should be still C/S_R. Simulation results show that the above arguments hold.

6 Numerical Examples and Simulation Results

In this section, we demonstrate the applicability of our queueing model and present analytical results of flow-based MPLS switch under the different label-setup policies and different ratio of the number of labels to IP flows (L/F). We also illustrate the trade-off between key system parameters. The system parameters used in numerical examples and simulations are shown as follows.

- Maximum window size : 32 kbytes (equivalent to 64 packets)
- IP packet size : exponentially distributed with a mean of 512 bytes
- Switch output rate : a mean rate of 10 Mbps, assuming exponential packet transmission time
- *Receiving_Node* rate : a mean rate of 10 Mbps, assuming exponential packet processing time
- Label-setup time : exponentially distributed with a mean of 100 ms
- Label-release time : exponentially distributed with a mean of 20 ms
- Label-release timer threshold (T_{rel}) : Erlang-k distributed in modeling, constant in simulation
- Label-setup timer threshold (T_{set}) : constant
- Round-trip delay : exponentially distributed
- Triggering threshold (m) : an integer ranging from 1 to 64
- The ratio of the number of labels to IP flows (L/F) : variable
- k : 100
- H : 20

We observe from Figures 5, 6, 8, and 9 that as T_{rel} increases, at certain T_{rel} value, the label-setup rate starts to approach zero and the channel utilization approaches a constant. Such T_{rel} is called $T_{rel,critical}$. In other words, to set $T_{rel} >> T_{rel,critical}$ does not improve performance and should not be considered.

	Throughput (Mbps)		
	Number of flows = 100		
L/F	$T_{rel} = 2$ ms	$T_{rel} = 10$ ms	$T_{rel} = 50$ ms
0.2	0.328835	0.308199	0.174467
0.4	0.510638	0.507765	0.348805
0.6	0.594050	0.612191	0.523117
0.8	0.632416	0.667602	0.697490
1	0.634040	0.673893	0.871007

Table 1: Steady-state throughput with $T_{set} = 50$ ms under 300 ms round-trip delay and 100 IP flows

	Throughput (Mbps)		
	Number of flows = 500		
L/F	$T_{rel} = 2$ ms	$T_{rel} = 10$ ms	$T_{rel} = 50$ ms
0.2	0.329032	0.308078	0.174377
0.4	0.510791	0.507869	0.348732
0.6	0.594143	0.612251	0.523089
0.8	0.633340	0.668186	0.697459
1	0.633845	0.673699	0.871021

Table 2: Steady-state throughput with $T_{set} = 50$ ms under 300 ms round-trip delay and 500 IP flows

	Label-setup rate (1/s)		
	Number of flows = 100		
L/F	$T_{rel} = 2$ ms	$T_{rel} = 10$ ms	$T_{rel} = 50$ ms
0.2	1.312428	1.061906	0.000937
0.4	2.743671	2.190284	0.001807
0.6	4.250141	3.378969	0.002784
0.8	5.633737	4.570052	0.003668
1	5.709475	4.750179	0.004637

Table 3: Label-setup rate with $T_{set} = 50$ ms under 300 ms round-trip delay and 100 IP flows

	Label-setup rate (1/s)		
	Number of flows = 500		
L/F	$T_{rel} = 2$ ms	$T_{rel} = 10$ ms	$T_{rel} = 50$ ms
0.2	1.313764	1.061982	0.000925
0.4	2.743164	2.190032	0.001832
0.6	4.250622	3.376883	0.002784
0.8	5.689289	4.595030	0.003696
1	5.711491	4.751262	0.004632

Table 4: Label-setup rate with $T_{set} = 50$ ms under 300 ms round-trip delay and 500 IP flows

	Channel utilization (%)		
	Number of flows = 100		
L/F	$T_{rel} = 2$ ms	$T_{rel} = 10$ ms	$T_{rel} = 50$ ms
0.2	55.049805	39.399108	8.729889
0.4	47.655322	34.137748	8.725639
0.6	40.576237	28.810385	8.723786
0.8	35.426307	24.570431	8.722656
1	35.174659	24.034908	8.715281

Table 5: Channel utilization with $T_{set} = 50$ ms under 300 ms round-trip delay and 100 IP flows

	Channel utilization (%)		
	Number of flows = 500		
L/F	$T_{rel} = 2$ ms	$T_{rel} = 10$ ms	$T_{rel} = 50$ ms
0.2	55.033187	39.371474	8.723329
0.4	47.644193	34.137796	8.723028
0.6	40.576779	28.815846	8.721822
0.8	35.224792	24.493997	8.722743
1	35.144803	24.034570	8.715945

Table 6: Channel utilization with $T_{set} = 50$ ms under 300 ms round-trip delay and 500 IP flows

In simulations or numerical calculation, the round-trip delay, T_{rel}, and T_{set} are key parameters. However, we also find that neither the number of IP flows nor the number of labels alone can determine system behavior. Instead, the L/F ratio plays a key role. In order to demonstrate the results are the same for the same L/F regardless of the number of IP flows, we run simulations with $T_{set} = 50$ ms under 300 ms round-trip delay and the different number of IP flows (100, 500). Tables 1–6 show this evidence.

In Figures 4–6, we plot the throughput, the channel utilization, and the label-setup rate as a function of T_{rel} under 300 ms round-trip delay and different L/F. From Figure 4, *the throughput increases with the increase of* T_{rel} if L/F equals 1. This is because each flow can be allocated a label when requesting for a label individually. When L/F equals 0.2 and 0.4, the decrease of the throughput with the increase of T_{rel} is that most flows need to wait when requesting for a label. One can note that the number of labels equals the number of IP flows can hold the labels forever when T_{rel} is exceedingly long. This explains why the throughput, the label-setup rate, and the channel utilization remain unchanged when T_{rel} is above 45 ms, as observed in Figures 4, 5, and 6. Therefore, the system should approximate the topology-driven MPLS switch when T_{rel} is well above 45 ms, and the system model is close to the traffic-driven MPLS switch when T_{rel} is well below 45 ms. With these results, we also observe the following. When L/F is low (≤ 0.4), one should select the traffic-driven system because this will give the high throughput, and the high channel utilization in spite of suffering higher label-setup rate. If the throughput, and the channel utilization are of concerns, lower T_{rel} is preferred. If one needs lower label-setup processing overhead, higher T_{rel} is desired. Choosing the topology-driven system when L/F equals 1 will lead to high throughput and label processing overhead, but the channel utilization is low. Conversely, selecting the traffic-driven system will give high channel utilization, but harm the throughput and label processing overhead.

In Figures 7–9, we plot the throughput, the channel utilization, and the label-setup rate as a function of T_{rel} under 50 ms round-trip delay and different L/F. In this circumstances, the system behavior is approximately equivalent to the topology-driven MPLS switch when T_{rel} is well above 8 ms and is close to the traffic-driven MPLS switch when T_{rel} is well below 8 ms. The reason that the throughput decreases with decrease of T_{rel} is that the packets go back to the transmitter quickly, but label is not readily available.

When we use discrete-event simulations to simulate our queueing model, both T_{set} and T_{rel} are set to be constant. Although T_{rel} in the analytical model is Erlang-k distributed, it is extremely close to a constant timer with $k = 100$. The setup timer T_{set} in the analytical model is approximated by a pseudo threshold n and is set to a little larger than the T_{set} used in the simulation. This approximation technique is validated by simulation results shown in Figures 4–9.

7 Conclusions

The queueing model for an MPLS switch supporting both traffic-driven and topology-driven connection setup has been developed, and the procedures to analytically derive performance metrics are also presented. Via our

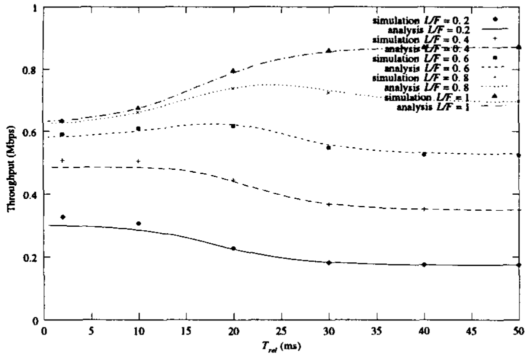

Figure 4: Throughput with T_{set} = 50 ms under 300 ms round-trip delay and different L/F.

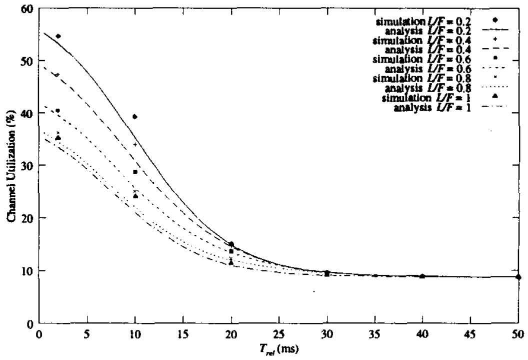

Figure 6: Channel Utilization with T_{set} = 50 ms under 300 ms round-trip delay and different L/F.

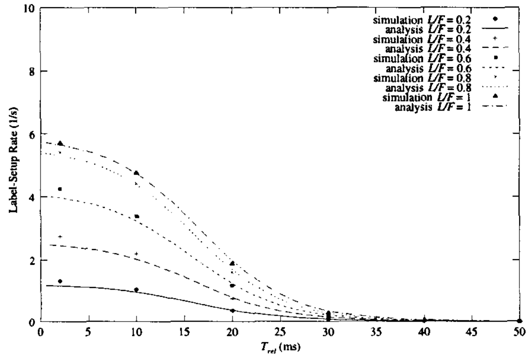

Figure 5: Label-setup rate with T_{set} = 50 ms under 300 ms round-trip delay and different L/F.

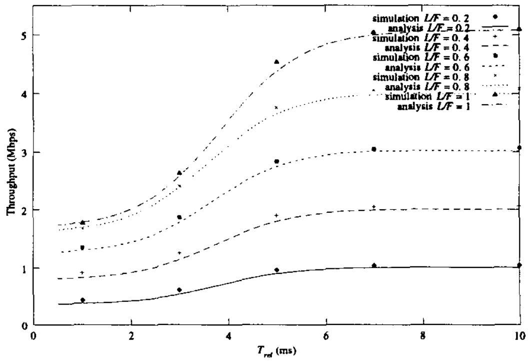

Figure 7: Throughput with T_{set} = 50 ms under 50 ms round-trip delay and different L/F.

proposed model, one can clearly observe the competition of multiple IP flow for limited resources. Although detailed operation of the label-setup and releases policies are also included in the queueing model. With only very limited computational complexity, the proposed mathematical model and the estimation procedures of label vacation time can quickly solve the desired performance metrics, such as the throughput, the label-setup rate, and the channel utilization.

In our previous work [16], observed from either simulations or analytical results, one can find that the attempt to achieve the high channel utilization and the high throughput at the same time is a conflict in the single IP flow environment. In the multiple IP flow model, this phenomenon is observed again when each flow has reserved its own label. In turn, in the multiple flow/multiple label operation, it is possible to achieve the high channel utilization and the high throughput in certain cases. With our model, the trade-off issues can be carefully examined so that network engineers can reach a balance between the channel bandwidth utilization and the system throughput performance. With an appropriate label-setup policy, one can achieve higher channel utilization with the minimal label processing overhead by selecting appropriate timer thresholds. Alternatively, one can predict the required processing overhead under performance constraints. Furthermore, our proposed queueing model also could be applied to the latest proposed "λ-labeling" MPLS system [7] since the resource under competition is simply replaced by λ.

In this paper, we have only analyzed the performance of a flow-based MPLS switch for multiple IP flow, assuming fixed window sizes and long-duration traffic. How to devise a sophisticated model including both short-duration and long-duration traffic, and how to illustrate the impact of variable-size windows of TCP on label competition should be an interesting area for future study.

References

[1] E. Rosen, A. Viswanathan, and R. Callon, "Multiprotocol Label Switching Architecture," *IETF Internet draft-ietf-mpls-arch-06.txt,* Aug. 1999.

[2] R. Callon, P. Doolan, N. Feldman, A. Fredette, and G. Swallow, "A Framework for Multiprotocol Label Switching," *IETF Internet draft-ietf-mpls-framework-05.txt,* Sept. 1999.

[3] D. Awduche, et al., "Requirements for Traffic Engineering over MPLS," *RFC 2702,* Sept. 1999.

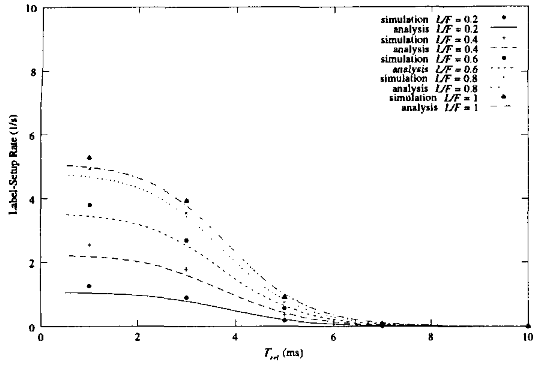

Figure 8: Label-setup rate with T_{set} = 50 ms under 50 ms round-trip delay and different L/F.

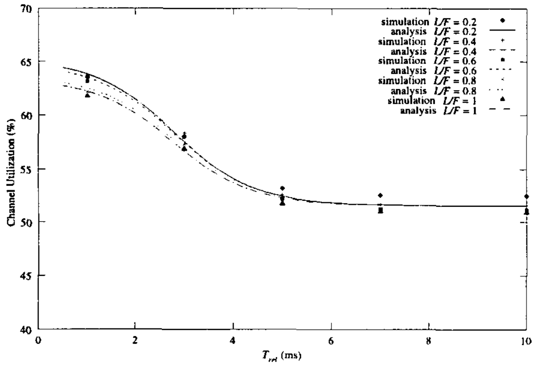

Figure 9: Channel Utilization with T_{set} = 50 ms under 50 ms round-trip delay and different L/F.

[4] X. Xiao, A. Hannan, and B. Bailey, "Traffic Engineering with MPLS in the Internet," *IEEE Network,* vol. 14, no. 2, pp. 28–33, March/April 2000.

[5] K. Nagami, H. Esaki, "Fow Aggregated, Traffic Driven Label Mapping in Label-Switching Networks," *IEEE J. Select. Areas Commun.,* vol. 17, no. 6, pp. 1170–1189, June 1999.

[6] S. Lin and N. Mckeown, "A simulation study of IP switching," *ACM SIGCOMM'97,* pp. 15–24, Palais des Festivals, Cannes, France.

[7] N. Ghani, "Lambda-Labeling: A Framework for IP-Over-WDM Using MPLS," *Workshop on Optical Networks,* Dallas, TX, Jan. 2000.

[8] P. Newman, G. Minshall, and T. Lyon, "IP Switching: ATM under IP," *IEEE/ACM Trans. Networking,* pp. 117–129, Aug. 1997.

[9] Z. Niu, Y. Takahashi, and N. Endo, "Performance Evaluation of SVC-Based IP-Over-ATM Networks," *IEICE Trans. Commun.,* vol. E81-B, no. 5, pp. 948–957, May 1998.

[10] Z. Niu and Y. Takahashi, "A Finite-Capacity Queue with Exhaustive Vacation/Close-Down/Setup Times and Markovian Arrival Processes," *Queueing Systems 31,* pp. 1–23, 1999.

[11] J. Zheng and V. O. K. Li, "Performance Model for IP Switch," *IEE Electronics Letters,* vol. 34, no. 21, pp. 2010–2012, Oct. 1998.

[12] S. Nakazawa, K. Kawahara, S. Yamaguchi, and Y. OIE, "Performance Comparasion with Layer 3 Switches in Case of Flow-And Topology-Driven Connection Setup," *IEEE GLOBECOM'99,* pp. 79–86, Rio de Janeiro, Brazil.

[13] P. G. Harrison and N. M. Patel, *Performance Modeling of Communication Networks and Computer Architectures,* pp. 288–295, Addison Wesley, 1993.

[14] K. M. Chandy, U. Herzog, and L. S. Woo, "Parameter Analysis of Queueing Networks," *IBM Journal of Research and Development,* vol. 19, no. 1, pp. 36–42, Jan. 1975.

[15] F. Baskett, K. M. Chandy, R. R. Muntz, and F. G. Palacios, "Open, Closed and Mixed Networks of Queues with Different Classes of Customers," *Journal of ACM,* vol. 22, no. 2, pp. 248–260, April 1975.

[16] L. C. Kao and Z. Tsai, "Performance Analysis of Flow-Based Label Switching: the Single IP Flow Model," to appear in *IEICE Trans. Commun.,* July 2000.