PAPER

Steady-State Performance Analysis of MPLS Label Switching*

Ling-Chih KAO† ***and*** **Zsehong TSAI**††**, *Nonmembers***

SUMMARY In this paper we propose a close-loop queueing model of MPLS switch under 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 number of 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, 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 works on related engineering or modeling.

For example, S. Lin and et al. in [6] studied the VC space requirement for different flow classifier and timer values in layer 2/3 switching system, and their method also could be applied to MPLS system. More related research works in this area were on the mathematical model of generic flow-based label switches or IP switches, such as [9]–[14], which also covers Ethernet-based or ATM-based switches. In [9]–[11], Niu and et al. proposed a performance model for SVC-based IP-over-ATM networks. In [12], they also studied the impact of the burstiness of the input traffic on the performance of SVC-based IP-over-ATM networks. In [13], Zheng and Li developed another performance model of IP/ATM switch. Their model could be used to evaluate the percentage of flows switched and the required virtual channel space size in an IP/ATM switch. In [14], Nakazawa and et al. presented a mathematical model with which both traffic-driven and topology-driven IP switches can be evaluated.

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 [19] 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], [19] are approximately similar under heavy traffic loads. The label-release policy is discussed more frequently (see [9]–[12], [19]). The label can be released only during the maximum allowed inactive duration, if no packets arrive. Our work [19]

Manuscript received September 1, 2000.
Manuscript revised February 23, 2001.

†The author is with the Graduate Institute of Communication Engineering, National Taiwan University, Taipei, Taiwan, R.O.C.

††The author is with the Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan, R.O.C.

*This work was supported by National Science Council, R.O.C., under Grant NSC 89-2213-E-002-168, and by Ministry of Education, R.O.C., under Grant 89E-FA06-2-4-7. An earlier version of this paper was published in part in ICOIN-15, Beppu-shi, Japan.

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 [13], [14] 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 [19] 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 models. Since it is difficult to directly manipulate an exact but often 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 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 Sect. 2, the queueing model for a flow-based MPLS switch is described. In Sect. 3, an approximate and efficient analysis is presented. In Sect. 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 Sect. 5. Numerical examples and simulation results are presented in Sect. 6. Conclusions are drawn in Sect. 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, where $N \geq M$. Since the objective is to model the behavior of flow-based MPLS switching under heavy traffic loads, the employed model is a close-loop queueing network model. The same fixed window size for each flow is assumed.

The detailed queueing network, called the original model, is shown in Fig. 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 message 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 a 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.

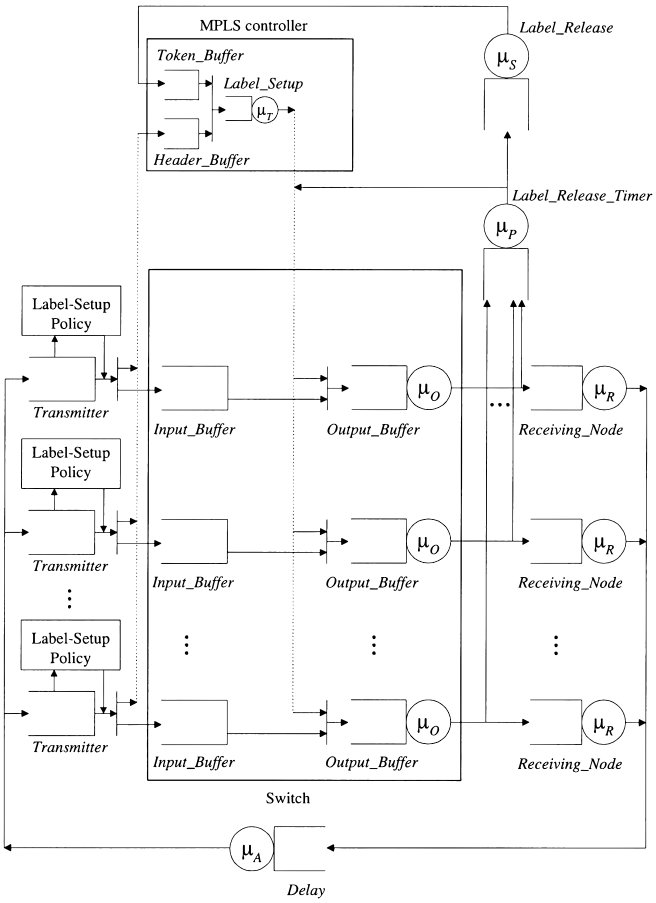


Fig. 1 Original queueing model.

When a burst of packets for a flow arrives, 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 the *Input_Buffer* of the switch. If the condition that a token resides in the *Token_Buffer* and the accumulated packets which reside in the *Input_Buffer* of the switch have 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 a 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 ($N = M$) is that a flow can get a label immediately when it requests for a label. Concerning this part, one can refer to [19] 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 multiple IP flow can be modeled via a label *vacation* node, if the label vacation time can be estimated accurately. As a result, the steady-state queueing model in Fig. 2 is proposed for performance analysis.

In this model, we 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.

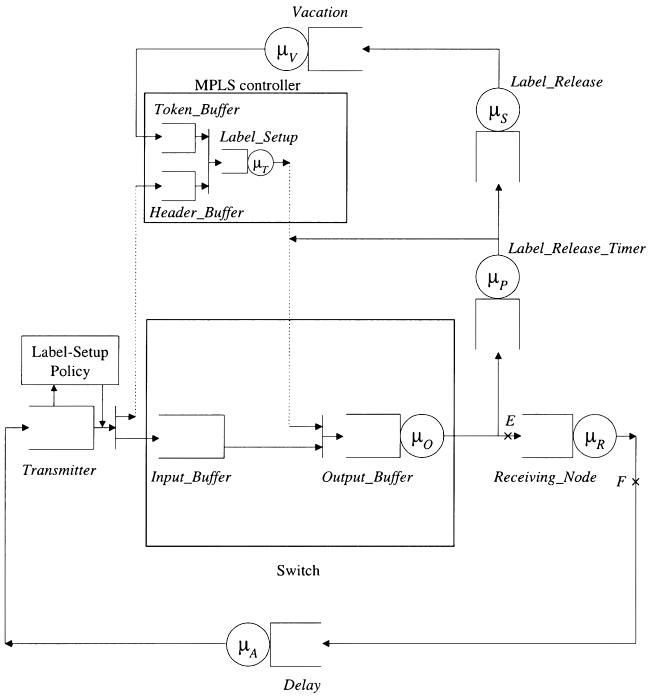


Fig. 2 Steady-state queueing model.

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 Fig. 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.
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 minimum number of accumulated packets that will trigger label-setup operation, ranging from 1 to W.

Unfortunately, the steady-state queueing model shown in Fig. 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 Fig. 2 so that the equivalent Norton's theorem in queueing analysis (see [15, p.250] or

[16]) 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 Fig. 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.

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

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

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

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

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

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

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

$$H\mu_V P_{0,i,0,1,0,j+1} + (i+1)\mu_A P_{0,i+1,0,1,0,j} = (i\mu_A + H\mu_V) P_{0,i,0,1,0,j},\ 0 \le i \le W-1,\ 1 \le j \le H-1 \tag{8}$$

$$(i+1)\mu_A P_{0,i+1,0,1,0,0} = (\mu_S + i\mu_A) P_{0,i,0,1,0,0}, \quad 0 \le i \le W-1 \tag{9}$$

$$(i+1)\mu_A P_{0,i+1,0,0,1,0} + H\mu_V P_{0,i,0,1,0,1} = (i\mu_A + \mu_T) P_{0,i,0,0,1,0}, \quad 0 \le i \le W - min(m,n) \tag{10}$$

$$(i+1)\mu_A P_{0,i+1,0,0,1,0} + H\mu_V P_{0,i,0,1,0,1} = i\mu_A P_{0,i,0,0,1,0}, \quad W - min(m,n) + 1 \le i \le W-1 \tag{11}$$

$$(\mu_O + j\mu_A) P_{W-j,j,0,0,0,0} = \mu_T P_{0,j,0,0,1,0} + \mu_O P_{W-j+1,j-1,0,0,0,0} + (j+1)\mu_A \cdot P_{W-j-1,j+1,0,0,0,0}, \quad 1 \le j \le W - min(m,n) \tag{12}$$

$$(\mu_O + j\mu_A) P_{W-j,j,0,0,0,0} = \mu_O P_{W-j+1,j-1,0,0,0,0} + (j+1)\mu_A P_{W-j-1,j+1,0,0,0,0}, \quad W - min(m,n) + 1 \le j \le W-2 \tag{13}$$

$$\mu_O P_{W,0,0,0,0,0} = \mu_T P_{0,0,0,0,1,0} + \mu_A P_{W-1,1,0,0,0,0} \tag{14}$$

where $P_{a,b,c,d,e,f}$ is the steady-state probability of the state vector (a, b, c, d, e, f). In Eqs. (10)–(13), n is the integer such that $\frac{1}{W\mu_A} + \frac{1}{(W-1)\mu_A} + ... + \frac{1}{(W-n+1)\mu_A}$ 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 Eqs. (1)–(14) is shown in Fig. 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 [15]. 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 Eqs. (1)–(14) effectively even for large state space. The details of this algorithm are shown as follows.

step 1: $P_{0,W,1,0,0,0} = 1$;

step 2: $P_{0,W,0,1,0,0} = \dfrac{k\mu_P}{\mu_S + W\mu_A} P_{0,W,1,0,0,0}$;

stept 3: $P_{0,W,0,1,0,H} = \dfrac{\mu_S}{H\mu_V + W\mu_A} P_{0,W,0,1,0,0}$;

step 4: $for\ (i = H-1;\ i \ge 1;\ i--)$

$$P_{0,W,0,1,0,i} = \frac{H\mu_V}{H\mu_V + W\mu_A} P_{0,W,0,1,0,i+1};$$

step 5: $P_{0,W,0,0,1,0} = \dfrac{H\mu_V}{W\mu_A} P_{0,W,0,1,0,1}$;

step 6: $for\ (i = 1;\ i \le k-1;\ i++)$

$$P_{0,W,i+1,0,0,0} = \left(1 + \frac{W\mu_A}{k\mu_P}\right) P_{0,W,i,0,0,0};$$

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

step 7: $P_{1,W-1,0,0,0,0} = \dfrac{k\mu_P + W\mu_A}{\mu_O} P_{0,W,k,0,0,0};$

step 8: $for\ (i = W-1;\ i \geq 0;\ i--)$

$$P_{0,i,0,1,0,0} = \frac{(i+1)\mu_A}{i\mu_A + \mu_S} P_{0,i+1,0,1,0,0};$$

step 9: $for\ (i = W-1;\ i \geq 0;\ i--)$

$$P_{0,i,0,1,0,H} = \frac{(i+1)\mu_A}{i\mu_A + H\mu_V} P_{0,i+1,0,1,0,H} + \frac{\mu_S}{i\mu_A + H\mu_V} P_{0,i,0,1,0,0};$$

step 10: $for\ (j = H-1;\ j \geq 1;\ j--)$

$for\ (i = W-1;\ i \geq 0;\ i--)$

$$P_{0,i,0,1,0,j} = \frac{H\mu_V}{H\mu_V + i\mu_A} \cdot P_{0,i,0,1,0,j+1} + \frac{(i+1)\mu_A}{H\mu_V + i\mu_A} P_{0,i+1,0,1,0,j};$$

step 11: $for\ (i = W-1;\ i \geq W - min(m,n) + 1;\ i--)$

$$P_{0,i,0,0,1,0} = (1 + \frac{1}{i}) P_{0,i+1,0,0,1,0} + \frac{H\mu_V}{i\mu_A} P_{0,i,0,1,0,1};$$

step 12: $for\ (i = W - min(m,n);\ i \geq 0;\ i--)$

$$P_{0,i,0,0,1,0} = \frac{(i+1)\mu_A}{i\mu_A + \mu_T} P_{0,i+1,0,0,1,0} + \frac{H\mu_V}{i\mu_A + \mu_T} P_{0,i,0,1,0,1};$$

step 13: $if\ (min(m,n) = 1)$

$for\ (2 \leq i \leq W-1)$

$$P_{i,W-i,0,0,0,0} = \frac{\mu_T}{\mu_O} \sum_{j=0}^{W-i} P_{0,j,0,0,1,0} + (W-i+1)\frac{\mu_A}{\mu_O} \cdot P_{i-1,W-i+1,0,0,0,0};$$

$else$

$for\ (2 \leq i \leq min(m,n) - 1)$

$$P_{i,W-i,0,0,0,0} = \frac{\mu_T}{\mu_O} \sum_{j=0}^{W-min(m,n)} P_{0,j,0,0,1,0}$$

$$+(W-i+1)\frac{\mu_A}{\mu_O} \cdot P_{i-1,W-i+1,0,0,0,0};$$

$$for\ (min(m,n) \le i \le W-1)$$

$$P_{i,W-i,0,0,0,0} = \frac{\mu_T}{\mu_O}\sum_{j=0}^{W-i} P_{0,j,0,0,1,0} + (W-i+1)\frac{\mu_A}{\mu_O} + P_{i-1,W-i+1,0,0,0,0};$$

step14: $P_{W,0,0,0,0,0} = \frac{\mu_T}{\mu_O}P_{0,0,0,0,1,0} + \frac{\mu_A}{\mu_O}P_{W-1,1,0,0,0,0};$

step 15: Normalize $P_{i,j,k,l,m,n}$

In this procedure, step 13 is based on aggregation of (12) and (13). 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:

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

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 [17], [18].

When we calculate the throughput of the equivalent queueing model, the service rate of *equivalent* queue must be set equal to $\eta(W-x)$ when the queue size of the *Receiving_Node* is equal to x. Owing to our label-setup policy, $\eta(W-x)$ is set to $\eta(min(m,n))$ when $W - x \le min(m,n)$.

4. Performance Measures

The steady-state throughput, the fairness index, 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 fairness index is about the fairness of the label usage. The label-setup rate is proportional to the required label processing load. In other words, the higher the label-setup rate, the heavier the label processing load. With the channel utilization, one can predict the ratio of wasted bandwidth. The formulas 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

$$T_C = \frac{WP_{size}}{\eta^\star(W)} \quad (16)$$

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

The throughput formula for $\eta^\star(W)$ is given by

$$\eta^\star(W) = \mu_R\left(1 - 1/\left(1 + \sum_{i=1}^{W}\eta(W) \cdot \eta(max(W-1, min(m,n))) \cdots \cdot \eta(max(W-i+1, min(m,n)))/\mu_R^i\right)\right) \quad (17)$$

The derivation of Eq. (17) can be referred to Eq. (10) of Ref. [19].

4.2 Fairness Index

The fairness index FRI is defined as

$$FRI = \sum_{i=1}^{N}\left(\frac{\eta^\star(W)_{i,actual} - \eta^\star(W)_{ave}}{\eta^\star(W)_{ave}}\right)^2 \quad (18)$$

where $\eta^\star(W)_{i,actual}$ is the steady-state throughput that the flow i actually obtains under a given R (the ratio of the number of labels to the number of IP flows), and $\eta^\star(W)_{ave}$ is the average steady-state throughput per flow obtains under the same R, which equals $\sum_{i=1}^{N}\frac{\eta^\star(W)_{i,actual}}{N}$.

4.3 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

$$S_R \approx \mu_T \sum_{i=0}^{W-min(m,n)} P_{0,i,0,0,1,0} \quad (19)$$

Here, we assume the capacity of the *Receiving_Node* is much larger than $\eta(W)$. If μ_R is not much larger than $\eta(W)$, Eq. (19) still serves as a worse case (upper bound) approximation.

4.4 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

$$U_R \approx \sum_{i=1}^{W} P_{i,W-i,0,0,0,0} \Big/ \left(\sum_{i=1}^{W} P_{i,W-i,0,0,0,0} + \sum_{i=1}^{k} P_{0,W,i,0,0,0} + \sum_{i=0}^{W} P_{0,i,0,1,0,0} \right) \quad (20)$$

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 $\eta(W)$, Eq. (20) 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 [19] or from the model of this paper (via Eq. (19)) with negligible vacation time. Before obtaining 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 labels for use immediately, while others probably wait for 1 or more cycles of token rotation. Therefore, we can approximate the waiting time (T_w) at *Label_Setup* node by estimating its lower bound, denoted by $T_{w,lowerbound}$, and its upper bound, denoted by $T_{w,upperbound}$, under this assumption by the following equations

$$T_{w,lowerbound} = \sum_{i=1}^{\lfloor \frac{N}{M} \rfloor - 1} iM/NS_R + \left(N - M \left\lfloor \frac{N}{M} \right\rfloor \right) \cdot \left(\left\lceil \frac{N}{M} \right\rceil - 1 \right) /NS_R \quad (21)$$

$$T_{w,upperbound} = \left(\left\lceil \frac{N}{M} \right\rceil - 1 \right) /S_R \quad (22)$$

Equations (21) and (22) can be illustrated by a simple example with 105 flows and 20 labels. In the worse case, 20 flows need to wait for one cycle of token rotation, 20 flows 2 cycles, 20 flows 3 cycles, 20 flows 4 cycles, and 5 flows 5 cycles. Taking average of these cycles of token rotation under arbitrary number of flows and labels gives rise to Eq. (21). The second term of Eq. (21) disappears as N is an integer multiple of M. Equation (22) provides the maximum possible value for expected waiting time.

Secondly, one can study the extreme case when T_{rel} is exceedingly long. In this case, the number of labels equals that of IP flows which 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 $(M/N)\eta^\star(W)_{|R=1}$, where R is the ratio of the number of labels to the number of IP flows and $\eta^\star(W)_{|R=1}$ is the steady-state throughput given that R equals 1 (which equals $\eta^\star(W)$ in [19] or Eq. (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 R: $\eta^\star(W)_{extreme} = (M/N)\eta^\star(W)_{|R=1}$, where $\eta^\star(W)_{extreme}$ is the throughput when T_{rel} is set close to infinite.

Finally, one can select an estimate value of $T_w \in \{T_{w,lowerbound}, T_{w,upperbound}\}$ which satisfies $|\eta^\star(W)_{|T_w} - \eta^\star(W)_{extreme}| < \epsilon$, 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 Sect. 3 with μ_V for T_w^{-1}, and ϵ is a small value. The last criterion is based on a conservation law in a simple close-loop queueing network, as shown in Fig. 4, which approximates the behavior among N circulating customers (flows) constantly compete for M servers (labels). The conservation law as the rate of intermediate node $\mu_I \to \infty$ is that average cycle time normalized by average service time (μ^{-1}) is exactly M/N ($= R$). As a result, an approximate conservation law is observed: $S_R T_w = C$ in this system, where C is some constant. For different thresholds value of T_{rel} under the same given R, the waiting time for the next available label should be still C/S_R. Simulation results show that the above arguments hold.

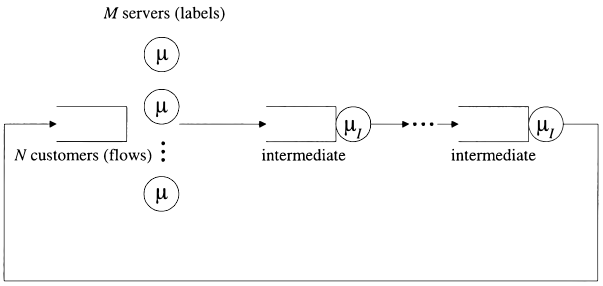


Fig. 4 An approximate close-loop queueing network illustrating N circulating customers (flows) constantly compete for M servers (labels).

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 = 100					
R	$T_{rel} = 2$ ms	$T_{rel} = 10$ ms	$T_{rel} = 20$ ms	$T_{rel} = 30$ ms	$T_{rel} = 40$ ms	$T_{rel} = 50$ ms
0.2	0.328835	0.308199	0.228754	0.182970	0.175408	0.174467
0.4	0.510638	0.507765	0.447365	0.365745	0.350680	0.348805
0.6	0.594050	0.612191	0.623574	0.548666	0.525915	0.523117
0.8	0.632416	0.667602	0.742862	0.728359	0.701297	0.697490
1	0.634040	0.673893	0.793410	0.858520	0.869602	0.871007

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

	Throughput (Mbps)					
	Number of flows = 500					
R	$T_{rel} = 2$ ms	$T_{rel} = 10$ ms	$T_{rel} = 20$ ms	$T_{rel} = 30$ ms	$T_{rel} = 40$ ms	$T_{rel} = 50$ ms
0.2	0.329032	0.308078	0.228578	0.182865	0.175314	0.174377
0.4	0.510791	0.507869	0.447570	0.365715	0.350587	0.348732
0.6	0.594143	0.612251	0.624135	0.548511	0.525885	0.523089
0.8	0.633340	0.668186	0.743417	0.728643	0.701223	0.697459
1	0.633845	0.673699	0.793220	0.858437	0.869653	0.871021

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 the number of IP flows (R). 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 (maximum TCP window size)
- The ratio of the number of labels to the number of IP flows (R): variable
- k: 100
- H: 20

We observe from Figs. 7, 8, 11, and 12 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 $T_{rel} \gg T_{rel,critical}$ does not improve performance and should not be considered.

In simulations or numerical calculations, 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, R plays a key role. In order to demonstrate the results are the same for the same R 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 the following discussions, if one is interested in how the number of labels affects the performance, the curves can be used by assuming the number of flows is fixed (N) but the number of labels is NR.

In Figs. 5–8, we plot the throughput, the fairness index, the channel utilization, and the label-setup rate as a function of T_{rel} with $T_{set} = 50$ ms under 300 ms round-trip delay and different R. From Fig. 5, the throughput increases with the increase of T_{rel} if R equals 1. This is because each flow can be allocated a label when requesting for a label. When R 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 individually. One can note that the number of

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 = 100					
R	$T_{rel} = 2$ ms	$T_{rel} = 10$ ms	$T_{rel} = 20$ ms	$T_{rel} = 30$ ms	$T_{rel} = 40$ ms	$T_{rel} = 50$ ms
0.2	1.312428	1.061906	0.391052	0.061824	0.007671	0.000937
0.4	2.743671	2.190284	0.781918	0.123658	0.015220	0.001807
0.6	4.250141	3.378969	1.184484	0.185893	0.022816	0.002784
0.8	5.633737	4.570052	1.597123	0.247259	0.030567	0.003668
1	5.709475	4.750179	1.874673	0.306735	0.038279	0.004637

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

	Label-setup rate (1/s)					
	Number of flows = 500					
R	$T_{rel} = 2$ ms	$T_{rel} = 10$ ms	$T_{rel} = 20$ ms	$T_{rel} = 30$ ms	$T_{rel} = 40$ ms	$T_{rel} = 50$ ms
0.2	1.313764	1.061982	0.390305	0.061777	0.007584	0.000925
0.4	2.743164	2.190032	0.782918	0.123557	0.015291	0.001832
0.6	4.250622	3.376883	1.184333	0.185335	0.022896	0.002784
0.8	5.689289	4.595030	1.598375	0.247612	0.030646	0.003696
1	5.711491	4.751262	1.875536	0.307284	0.038005	0.004632

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

	Channel utilization (%)					
	Number of flows = 100					
R	$T_{rel} = 2$ ms	$T_{rel} = 10$ ms	$T_{rel} = 20$ ms	$T_{rel} = 30$ ms	$T_{rel} = 40$ ms	$T_{rel} = 50$ ms
0.2	55.049805	39.399108	15.127407	9.544100	8.818179	8.729889
0.4	47.655322	34.137748	14.783928	9.538882	8.815088	8.725639
0.6	40.576237	28.810385	13.791370	9.539740	8.812186	8.723786
0.8	35.426307	24.570431	12.363284	9.497962	8.813650	8.722656
1	35.174659	24.034908	11.375979	9.060987	8.755125	8.715281

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

	Channel utilization (%)					
	Number of flows = 500					
R	$T_{rel} = 2$ ms	$T_{rel} = 10$ ms	$T_{rel} = 20$ ms	$T_{rel} = 30$ ms	$T_{rel} = 40$ ms	$T_{rel} = 50$ ms
0.2	55.033187	39.371474	15.101748	9.537625	8.811391	8.723329
0.4	47.644193	34.137796	14.798129	9.537393	8.811329	8.723028
0.6	40.576779	28.815846	13.795544	9.536423	8.811295	8.721822
0.8	35.224792	24.493997	12.371460	9.502415	8.812040	8.722743
1	35.144803	24.034570	11.375292	9.059081	8.754871	8.715945

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 Figs. 5, 7, and 8. 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 R is low (≤ 0.4), one should select the traffic-driven system because this will give high throughput, low fairness index, and high channel utilization in spite of suffering higher label-setup rate. If the throughput, the fairness index, 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. If one configures the switch into the topology-driven system by setting label-to-flow ratio

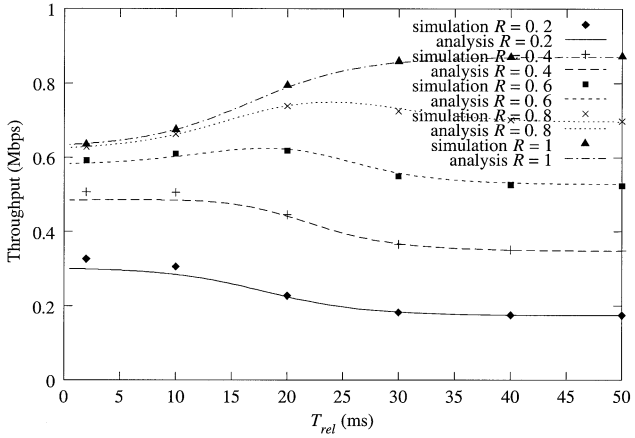


Fig. 5 Throughput with $T_{set} = 50$ ms under 300 ms round-trip delay and different R.

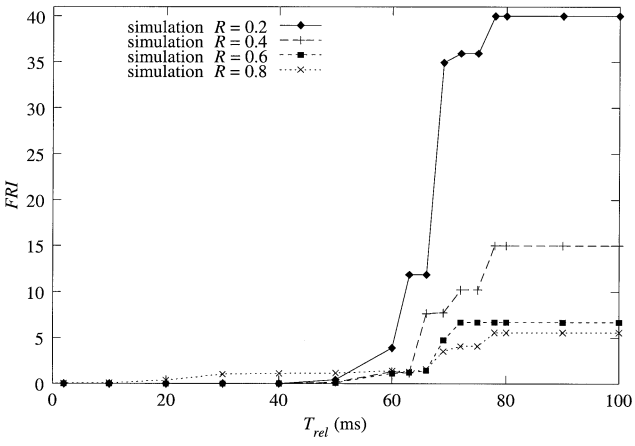


Fig. 6 Fairness index with $T_{set} = 50$ ms under 300 ms round-trip delay and different R.

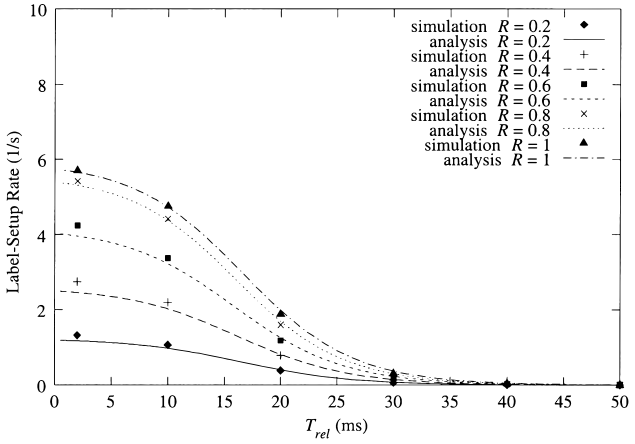


Fig. 7 Label-setup rate with $T_{set} = 50$ ms under 300 ms round-trip delay and different R.

$R = 1$ and selecting large T_{rel} will lead to high throughput and label processing overhead, but the channel utilization is low. Conversely, selecting the traffic-driven

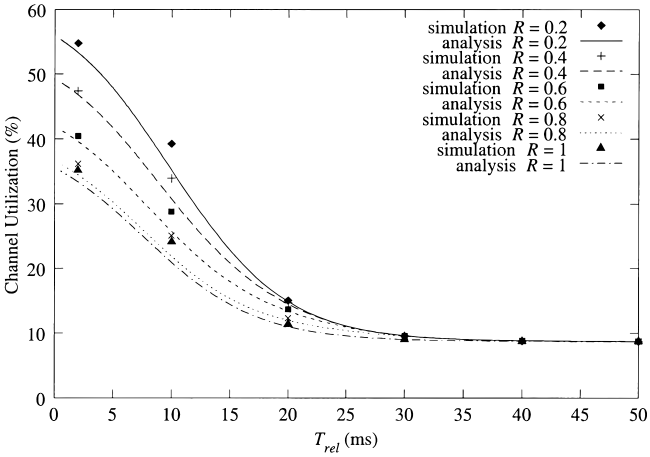


Fig. 8 Channel utilization with $T_{set} = 50$ ms under 300 ms round-trip delay and different R.

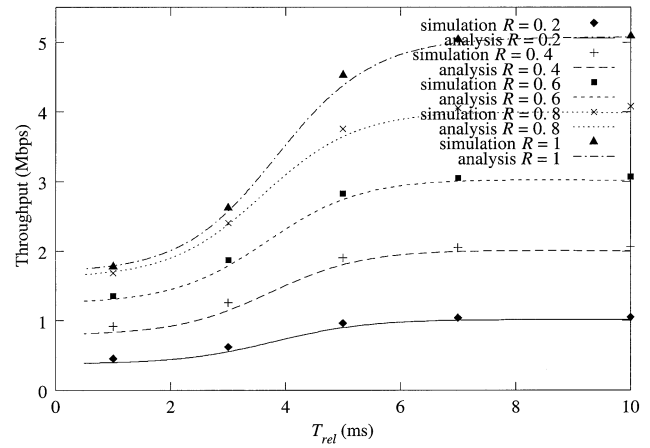


Fig. 9 Throughput with $T_{set} = 50$ ms under 50 ms round-trip delay and different R.

system will give high channel utilization, but harm the throughput and label processing overhead.

In Figs. 9–12, we plot the throughput, the fairness index, the channel utilization, and the label-setup rate as a function of T_{rel} with $T_{set} = 50$ ms under 50 ms round-trip delay and different R. 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} as $R < 1$ is that the packets go back to the transmitter quickly, but label is not readily available. The trade-off among the throughput, the label-setup rate, and the channel utilization is similar to the condition when $R = 1$ with $T_{set} = 50$ ms under 300 ms round-trip delay.

Another important observation can be made from Figs. 13–15. With large round-trip delay, efficient label-setup and release operations can easily increase the channel utilization. In turn, if the round-trip delay is

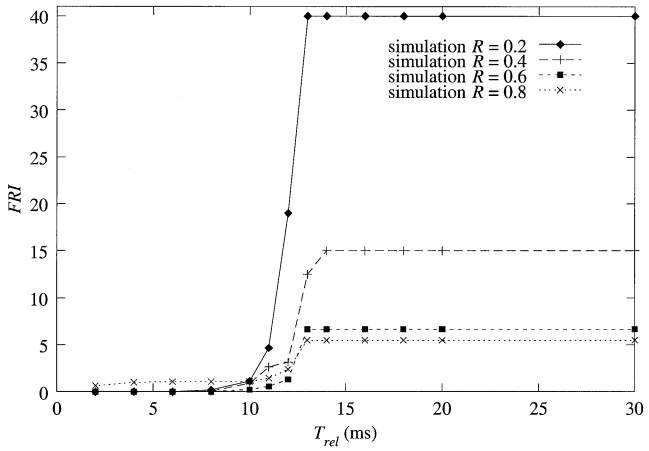


Fig. 10 Fairness index with $T_{set} = 50$ ms under 50 ms round-trip delay and different R.

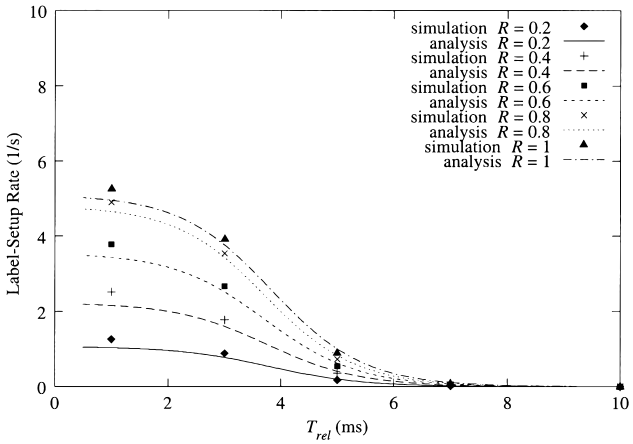


Fig. 11 Label-setup rate with $T_{set} = 50$ ms under 50 ms round-trip delay and different R.

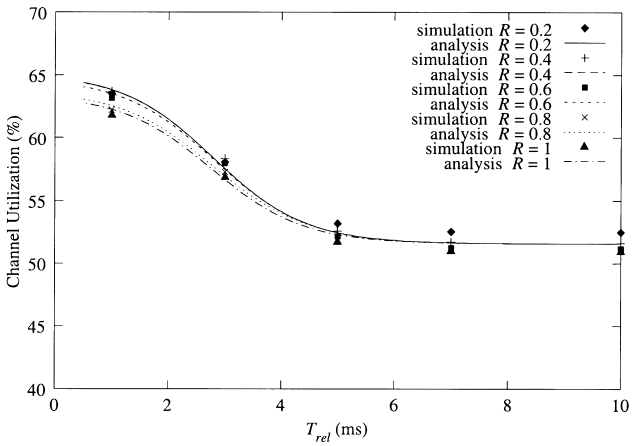


Fig. 12 Channel utilization with $T_{set} = 50$ ms under 50 ms round-trip delay and different R.

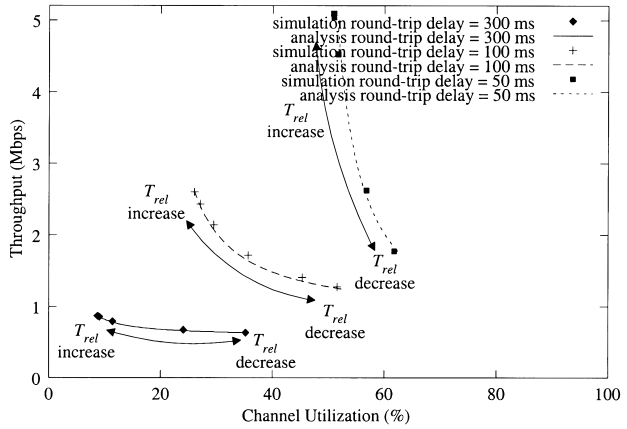


Fig. 13 Throughput versus channel utilization with $T_{set} = 50$ ms and $R = 1$.

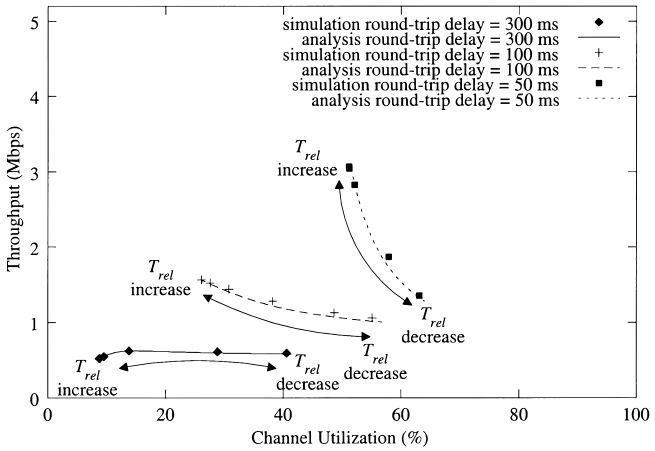


Fig. 14 Throughput versus channel utilization with $T_{set} = 50$ ms and $R = 0.6$.

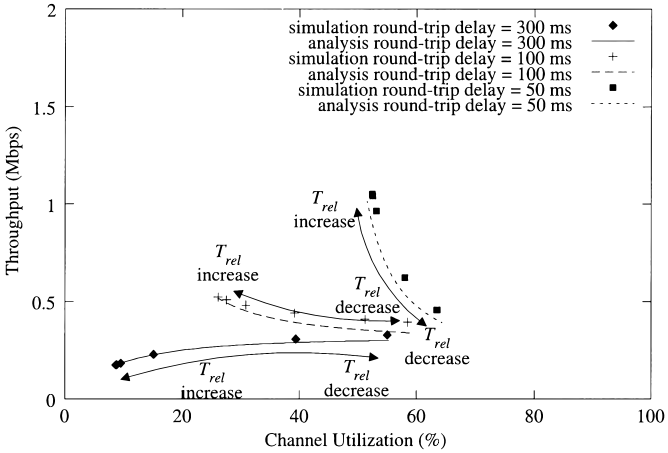


Fig. 15 Throughput versus channel utilization with $T_{set} = 50$ ms and $R = 0.2$.

small (≤ 50 ms), the improvement of utilization is less significant. In order to illustrate the trade-off between the throughput and the channel utilization by adjusting T_{rel} under various round-trip delays and R, we plot the throughput versus the channel utilization under various round-trip delays with a fixed T_{set} as shown in Figs. 13–

15. With these figures, one not only can easily determine T_{rel} according to the required throughput and the channel utilization, but also can clearly observe the improvement of channel utilization under various network environments.

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 $k = 100$. Similarly, the setup timer T_{set} in the analytical model is approximated by a pseudo constant threshold n and is set to be a little larger than the T_{set} used in the simulation. This approximation technique is validated by simulation results shown in Figs. 5–15.

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 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, 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.

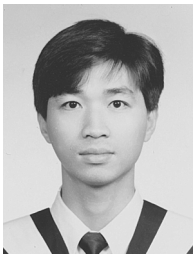
In our previous work [19], observed from either simulations or analytical results, one can find that the attempt to achieve high channel utilization and high throughput via label setup policy at the same time is a conflict in the single IP flow environment. With the multiple IP flow MPLS model, this phenomenon is reexamined again. In turn, in the multiple flow/multiple label operation, we find that it is possible to achieve high channel utilization and high throughput in many cases, but it requires detailed engineering. 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 also 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 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.

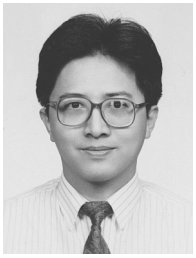
References

[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, J. Malcolm, J. Agogbua, M. O'Dell, and J. McManus, "Requirements for traffic engineering over MPLS," RFC 2702, Sept. 1999.
[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 and H. Esaki, "Fow aggregated, traffic driven label mapping in label-switching networks," IEEE J. Sel. 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, vol.6, no.2, pp.117–129, April 1998.
[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, "An extented queueing model for SVC-based IP-over-ATM networks and its analysis," IEEE GLOBECOM'98, pp.1950–1955, Sydney, Australia, 1998.
[11] 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.
[12] N. Endo, A. Sawada, and Z. Niu, "The effect of burst duration of input traffic in SVC-based IP-over-ATM networks," 11th ITC Specialist Seminar, pp.76–81, Yokohama, Japan, 1999.
[13] J. Zheng and V.O.K. Li, "Performance model for IP switch," IEE Electronics Letters, vol.34, no.21, pp.2010–2012, Oct. 1998.
[14] 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, 1999.
[15] P.G. Harrison and N.M. Patel, Performance Modeling of Communication Networks and Computer Architectures, pp.288–295, Addison Wesley, 1993.
[16] K.M. Chandy, U. Herzog, and L.S. Woo, "Parameter analysis of queueing networks," IBM J. Research and Development, vol.19, no.1, pp.36–42, Jan. 1975.
[17] F. Baskett, K.M. Chandy, R.R. Muntz, and F.G. Palacios, "Open, closed and mixed networks of queues with different classes of customers," J. ACM, vol.22, no.2, pp.248–260, April 1975.
[18] J. Mckenna, D. Mitra, and K.G. Ramakrishnan, "A class of closed Markovian queueing networks: Integral representations, asymptotic expansions, and generations," The Bell System Technical Journal, vol.60, pp.599–641, May–June 1981.

[19] L.C. Kao and Z. Tsai, "Performance analysis of flow-based label switching: The single IP flow model," IEICE Trans. Commun., vol.E83-B, no.7, pp.1417–1425, July 2000.



Ling-Chih Kao was born in Taiwan, R.O.C., on May 20, 1972. He received the B.S. and M.S. degrees in electrical engineering from Yuan Ze University, Taoyuan, Taiwan in 1995 and 1997. He is currently working toward the Ph.D. degree in electrical engineering at the National Taiwan University, Taipei, Taiwan. His research interests include queueing theory, teletraffic modeling, wireless communication protocols.



Zsehong Tsai received the B.S. degree in electrical engineering from National Taiwan University (NTU), Taipei, in 1983, and the M.S. and Ph.D. degrees from the University of California, Los Angeles, in 1985 and 1988, respectively. During 1988–1990, he worked as a Member of Technical Staff at AT&T Bell Laboratories, where he investigated performance aspects of network management systems. Since 1990, he has been with the Department of Electrical Engineering of NTU, where he is currently a professor. He is also with the Graduate Institute of Communication Engineering and the Graduate Institute of Industrial Engineering at NTU. Dr. Tsai was responsible for the technical support of the ATM testbed deployment and many ATM interoperability trials at National Taiwan University. Recently, he was assigned by National Science Council, R.O.C. as a technical coordinator responsible for the planning of Taiwan's National Broadband Experimental Network (NBEN). His research interests include high speed networking, broadband Internet, network management, network planning, and telecommunication (de)regulations. Dr. Tsai is a receipt of the CIE (Chinese Institute of Engineers) Technical Paper Award in 1997. He is also a member of IEEE.