PAPER

Performance Analysis of Flow-Based Label Switching: The Single IP Flow Model

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

SUMMARY A closed-loop queueing model of flow-based label switches, supporting label reservation protocols of different label-setup and release policies, is presented. This model can emulate the behavior of TCP under the label switch when the maximum window size has been achieved and the packet loss rate is negligible. The label-setup policy is that the IP controller does not start to set up a label until the accumulated packets of the same flow in the switch buffer have exceeded a triggering threshold. Meanwhile, the reserved bandwidth is released when the flow is detected idle and the label-release timer has expired. This policy can achieve higher channel utilization with minimal label processing overhead in spite of suffering from certain delay penalty. To avoid unnecessary TCP timeout or large packet delay under such policy, we also introduce a label-setup timer. Norton's theorem is applied to obtain approximate solutions of this queueing model. Although the analytical method is an approximate one, the simulation results show that the accuracy is high and this model can clearly illustrate how the label-setup and the lable-release timer affect the system performance. Besides, one can observe the trade-off between the throughput and the channel utilization.

key words: *flow, label switch, performance analysis*

1. Introduction

We consider a flow-based label switch in which the network bandwidth and labels can be dynamically reserved for each flow. A flow is defined as a sequence of packets that are treated identically by the complex routing function employed in flow-based label switching systems, where the label switch forwards IP packets using either IP address or label that is mapped to the IP address [7], [8]. The label switch is superior than 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 layer-2 switching. But directly combining the operations of layer-2 switching and layer-3 IP routing involves unnecessary overhead, namely, some functions are duplicate and certain bandwidth might be wasted. Therefore, it is necessary for a well-designed high-speed IP router to meet today's requirements by using efficient cut-through techniques. These new techniques take advantage of IP routing and high speed switching, while maintain high bandwidth utilization. They include classical IP over ATM and NHRP [1], Cisco's Tag Switch [2], Toshiba's Cell Switch Router (CSR) [3], Ipsilon's IP Switch [4], [5], [9], and IBM's Aggregate Route Based IP Switching (ARIS) [6]. In order to integrate these techniques, the Internet Engineering Task Force (IETF) recently proposes the MultiProtocol Label Switching (MPLS) architecture [7], [8] which can be used for transport of any level 3 protocol over any level 2 technology.

Most related research works in this area discuss the architectural issue of flow-based label switches or IP switches. From [9], we know that most Internet traffic processed by IP switches is long-lived. So our attention focus on the effects of long-duration traffic passing through a label switch. So far, only a few papers [10]–[15] explored the mathematical model of generic layer 2/3 switch, including Ethernet-based or ATM-based switches, under long-duration traffic sources. In [10]–[12], Niu and et al. proposed a performance model for SVC-based IP-over-ATM networks. However, they did not suggest how to reduce the processing overhead of IP controller, and how to determine an adequate label-release timer for increasing the channel utilization. In [13], they studied the impact of the burstiness of the input traffic on the performance of SVC-based IP-over-ATM networks. In [14], the important system design parameters, such as buffer size, bandwidth, and SVC timeout period, on the performance SVC-based IP-over-ATM networks were investigated. In [15], Zheng and Li developed another performance model of IP/ATM switch. Their model can be used to evaluate the percentage of flows switched and the required virtual channel space size in an IP/ATM switch. In the models proposed in previous research works, none of them have been used to evaluate the trade-off between the label-setup policy and the channel (link) utilization or throughput of IP streams. An efficient model for investigating the related performance issues and the trade-off in network engineering is thus strongly desired.

Our flow-based label switch queueing model supporting flow-based IP transport is modified from the SVC-based IP-over-ATM model proposed by [10]. First, we use closed-loop queueing network model with

Manuscript received August 25, 1999.

Manuscript revised November 25, 1999.

[†]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.

round-trip delay because our attention is on heavy traffic conditions, i.e., we assume the maximum TCP window size is maintained in the communication. Secondly, the IP controller (or label processor) does not start to set up a label until the switch has accumulated enough packets. The threshold for accumulated packets to trigger label-setup is a design parameter. Although our label-setup policy can increase the channel utilization and reduce the label processing overhead, a large triggering threshold may cause unnecessary TCP timeout or large packet delay. To overcome this problem, the label-setup timer is included to assure that packet delay does not exceed TCP timeout value. The label-setup policy is modified such 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. We not only investigate the effects of the label-setup timer and the label-release timer on a flow-based label switch, also obtain the trade-off between the throughput and the channel utilization. Besides, a set of different label-setup policies incorporating different thresholds can be devised. We believe it is possible to achieve the required performance under reasonable channel utilization by selecting an appropriate label-setup policy and release timer with our model.

The remainder of the paper is organized in the following. In Sect. 2, the queueing model for a flow-based label switch is described. In Sect. 3, an approximate and efficient analysis is presented. In Sect. 4, three performance measures are derived. Numerical examples and simulation results are presented in Sect. 5. Conclusions are drawn in Sect. 6.

2. Queueing Model

In this section, we present a queueing model which characterizes the behavior of the single IP flow on a flow-based label switch. Since our objective is to model the behavior of a flow-based label switch during heavy traffic loads, we employ a close-loop queueing network model with a finite window size W.

The original queueing network is shown in Fig. 1. Eleven queueing nodes in total are included in this model: *Token_Buffer, Header_Buffer, Label_Setup, Label_Release, Label_Release_Timer, Transmitter, Input_Buffer, Output_Buffer, Forward_Channel, Receiver* and *Ack_Channel.* In order to simplify the performance analysis, the *Forward_Channel* and the *Ack_Channel* are lumped into the *Delay* queue in the steady-state analysis. This should not affect the steady-state behavior of other queueing nodes. Because if one changes the location of the *Receiver* and the *Forward_Channel,* the throughput can be kept the same (see [16]). The steady-state queueing model for performance analysis is shown in Fig. 2. The *Transmitter,* the *Input_Buffer,* the *Output_Buffer,* the *Receiver* and the *Delay* queue

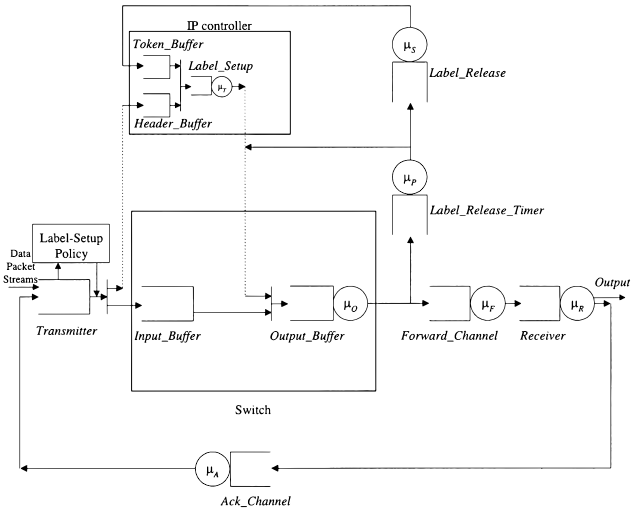


Fig. 1 Original queueing model.

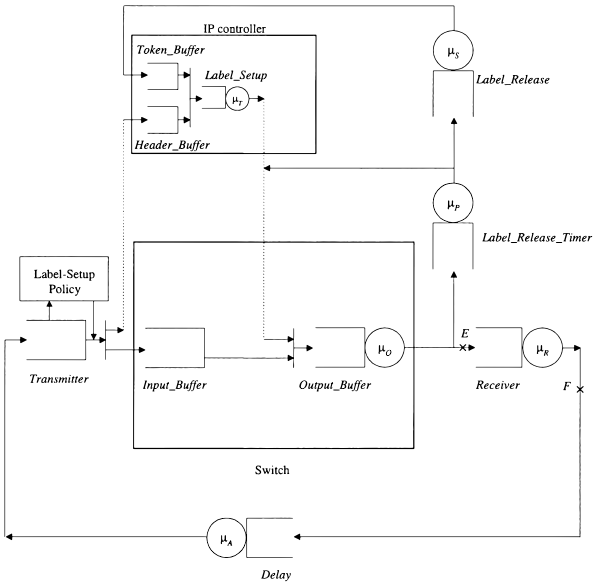


Fig. 2 Steady-state queueing model.

form the message loop. The *Token_Buffer* stores the token which represents 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 the flow. The *Label_Setup* queue represents the time required to look up the routing 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 inactive label-release timer, which is a timer indicating the length of idle period of a flow. The *Delay* queue represents the total round-trip delay of the forward and the feedback channels. The sending rate of the transmitter is assumed to be larger than the switch output and the receiver. The switch output is assumed to have enough buffers, so the packet loss rate is negligible. The de-

tailed operations are described as follows.

When a burst of packets arrive, the header information of the first packet is extracted and copied into the *Header_Buffer* of the IP controller. The leading packet and its following packets then enter into the *Input_Buffer* of the switch. If a token resides in the *Token_Buffer,* and the accumulated packets reside in the *Input_Buffer* of the switch have exceeded the triggering threshold or the label-setup timer has expired, the IP 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 inactive timer, controlling label-release time, to each label. The label is released only if no packets arrive during the maximum allowed inactive duration.

In our model, we use Erlang-k distribution in *Label_Release_Timer* to approximate a constant label-release timer and employ infinite server queue 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 label switch, considering only the single IP flow with a fixed window size W. 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 *Receiver* 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}: label-setup timer
μ_S: service rate in the *Label_Release* queue
μ_T: service rate in the *Label_Setup* 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
S_T: the current phase at which the token stays in the *Label_Release_Timer* queue, ranging from 1 to k
S_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
m: the triggering threshold which represents the number of accumulated packets waiting for setting up a label, ranging from 1 to W

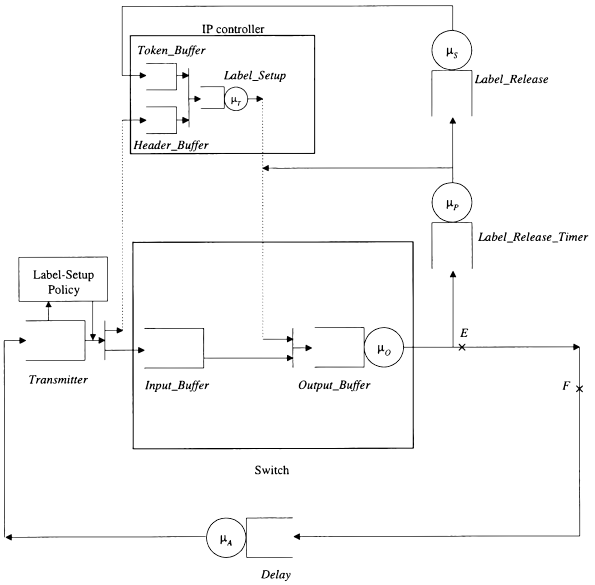


Fig. 3 Shorted-circuit queueing model.

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 [17, p.250] or [18]) 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 Fig. 3. We then employ the state vector (a, b, c, d, e) to represent the state of the shorted-circuit model when $n_O = a, n_A = b, S_T = c, S_R = d$ and $n_T = e$.

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

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

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

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

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

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

$$(i+1)\mu_A P_{0,i+1,0,0,1} + \mu_S P_{0,i,0,1,0} = (i\mu_A + \mu_T) \cdot P_{0,i,0,0,1}, \quad 0 \le i \le W - \min(m,n) \tag{6}$$

$$(i+1)\mu_A P_{0,i+1,0,0,1} + \mu_S P_{0,i,0,1,0} = i\mu_A P_{0,i,0,0,1}, \quad W - \min(m,n) + 1 \le i \le W-1 \tag{7}$$

$$[\mu_O + j\mu_A]\, P_{W-j,j,0,0,0} = \mu_T P_{0,j,0,0,1} + \mu_O P_{W-j+1,j-1,0,0} + (j+1)\mu_A P_{W-j-1,j+1,0,0,0}, \quad 1 \le j \le W - \min(m,n) \tag{8}$$

$$[\mu_O + j\mu_A]\, P_{W-j,j,0,0,0} = \mu_O P_{W-j+1,j-1,0,0} + (j+1)\mu_A P_{W-j-1,j+1,0,0,0}, \quad W - \min(m,n) + 1 \le j \le W-2 \tag{9}$$

$$\mu_O P_{W,0,0,0,0} = \mu_T P_{0,0,0,0,1} + \mu_A P_{W-1,1,0,0,0} \tag{10}$$

where $P_{a,b,c,d,e}$ is the steady-state probability of the state vector (a,b,c,d,e). In Eqs. (6)–(9), 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, in stead of directly modeling T_{set}, we use a corresponding pseudo threshold n to limit the holding time of packets. The state-transition diagram corresponding to Eqs. (1)–(10) is shown in Fig. 4. In most cases, we set $n > m$. Consequently, in the balance equations, the label-setup timer is dominated by a triggering threshold m such 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 [17]. 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)–(10) effectively even for large state space. The details of this algorithm are shown as follows.

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

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

step 3: $P_{0,W,0,0,1} = \dfrac{\mu_S}{W\mu_A} P_{0,W,0,1,0}$

step 4: *for* $(i = 1;\ i \le k-1;\ i++)$

$$P_{0,W,i+1,0,0} = \left(1.0 + \frac{W\mu_A}{k\mu_P}\right) P_{0,W,i,0,0}$$

step 5: $P_{1,W-1,0,0,0} = \dfrac{k\mu_P + W\mu_A}{\mu_O} P_{0,W,k,0,0}$

step 6: *for* $(i = W-1;\ i \ge 0;\ i--)$

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

step 7: *for* $(i = W-1;\ i \ge W - \min(m,n) + 1;$ $i--)$

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

step 8: *for* $(i = W - \min(m,n);\ i \ge 0;\ i--)$

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

step 9: *if* $(\min(m,n) = 1)$

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

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

else

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

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

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

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

step 10: $P_{W,0,0,0,0} = \dfrac{\mu_T}{\mu_O} P_{0,0,0,0,1} + \dfrac{\mu_A}{\mu_O} P_{W-1,1,0,0,0}$

step 11: Normalize $P_{i,i,k,l,m}$

In this procedure, step 9 is based on aggregation

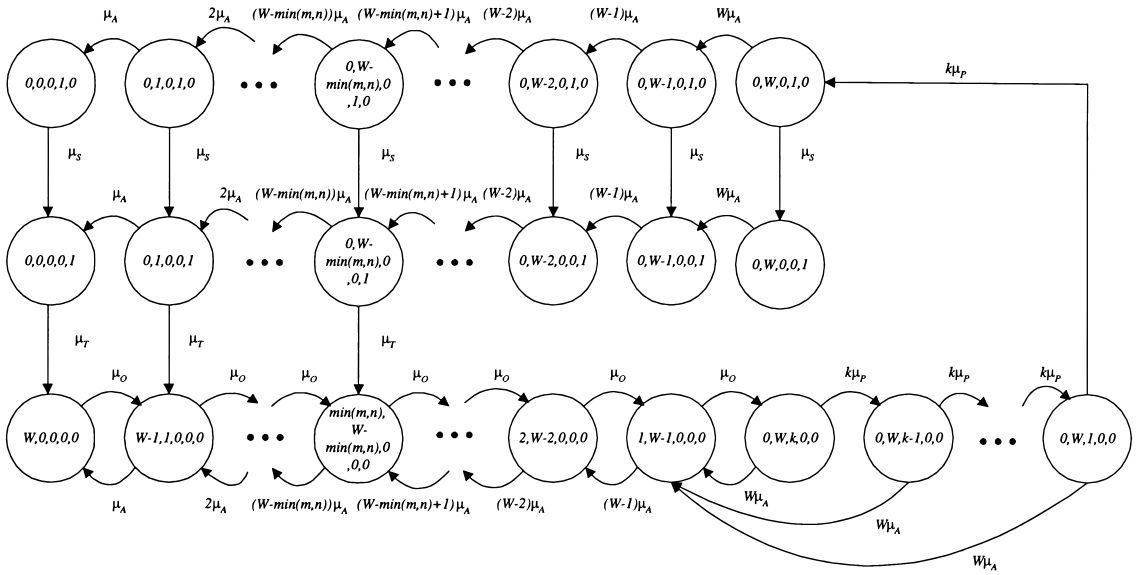


Fig. 4 State-transition diagram.

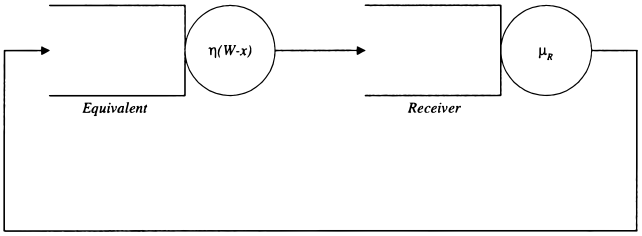


Fig. 5 Equivalent queueing network with state-dependent service rate at the *equivalent* server node, where x is the number of packets in the *Receiver* queue and W is the maximum window size.

of (8) and (9). Let $\eta(w)$ denote the throughput of the shorted-circuit queueing model, provided that the population is w. Then $\eta(w)$ can be calculated using the steady-state solution obtained via the above calculation procedure and the following equation:

$$\eta(w) = \mu_O \sum_{i=1}^{w} P_{i,w-i,0,0,0} \ , w = 1, 2, \ldots, W \qquad (11)$$

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 shown in Fig. 5.

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 [19], [20].

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 *Receiver* is equal to x. Owing to our label-setup policy, $\eta(W - x)$ is set to $\eta(\min(m, n))$ when $W - x \leq \min(m, n)$.

4. Performance Measures

The steady-state throughput, the label-setup rate and the channel utilization are considered as 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. These metrics are derived as follows.

4.1 Average Cycle Time and Throughput

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

$$T_C = \frac{WP_{size}}{\eta^\star(W)} \qquad (12)$$

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

$$\eta^\star(W) = \mu_R \Bigg(1 - 1/\Bigg(1 + \sum_{i=1}^{W} \eta(W) \cdot \eta(\max(W-1, \min(m,n))) \cdots \cdot \eta(\max(W-i+1, \min(m,n)))/\mu_R^i \Bigg) \Bigg) \qquad (13)$$

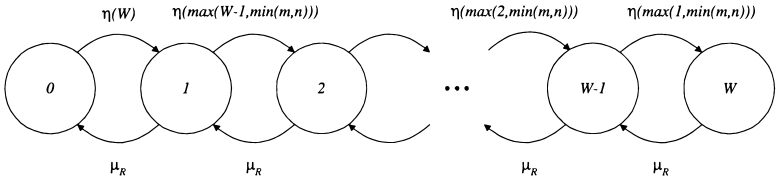


Fig. 6 State-transition diagram of the equivalent queueing network.

The derivation of Eq. (13) is as follows. Let the system state be the number of packets in the *Receiver* queue, and let π_i denote the steady-state probability, where i (ranging from $0 \sim W$) is the state. Since the corresponding process is a standard birth-and-death process and the service rate of the *equivalent* server node is limited to be larger or equal to $\eta(\min(m,n))$, the corresponding state-transition diagram of the equivalent queueing network is shown in Fig. 6. It can be shown that

$$\begin{aligned}\pi_i \;=\; & \eta(W)\eta(\max(W-1,\min(m,n)))\cdots \\ & \cdot\,\eta(\max(W-i+1,\min(m,n)))\pi_0/\mu_R^i\;, \\ & i=1,2,\ldots,W \end{aligned} \tag{14}$$

Due to $\sum_{i=0}^{W}\pi_i = 1$, and the fact that throughput can be determined by server utilization $1-\pi_0$, i.e.

$$\eta^\star(W) = \mu_R(1-\pi_0), \tag{15}$$

one can then derive (13).

4.2 Label-Setup Rate

The label-setup rate S_R is defined as the average number of label-setup operations in the IP 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

$$S_R \approx \mu_T \sum_{i=0}^{W-\min(m,n)} P_{0,i,0,0,1} \tag{16}$$

Here, we assume the capacity of the receiver is much larger than $\eta(W)$. If μ_R is not much larger than $\eta(W)$, Eq. (16) 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 switch serves packets to the time during the channel is reserved. It is given by

$$\begin{aligned}U_R \approx \sum_{i=1}^{W} P_{i,W-i,0,0,0} \Big/ \Bigg(& \sum_{i=1}^{W} P_{i,Wi,0,0,0} + \sum_{i=1}^{k} P_{0,W,i,0,0} \\ & + \sum_{i=0}^{W} P_{0,i,0,1,0} \Bigg)\end{aligned} \tag{17}$$

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 packet transmission in the channel is considered to be efficiently utilized. Here, we assume the capacity of the receiver is much larger than $\eta(W)$. If μ_R is not much larger than $\eta(W)$, Eq. (17) serves as a best case approximation.

5. Numerical Examples and Simulation Results

In this section, we demonstrate the applicability of our queueing model and present analytical results of flow-based label switches under different label-setup policies. 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
- Receiver 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
- k : 100

Among them, the round-trip delay, T_{rel}, and T_{set} are all key parameters. In Figs. 7–9, we plot the average cycle time, the channel utilization, and the label-setup rate as a function of T_{rel} under various round-trip delays. With Figs. 7 and 8, one can select an appropriate T_{rel} value when the channel utilization and the required throughput must be taken into account simultaneously. For example, if the target channel utilization is between 52% and 58% under 50 ms round-trip delay, then the required T_{rel} value is about 1.7 ms to 5 ms observed from

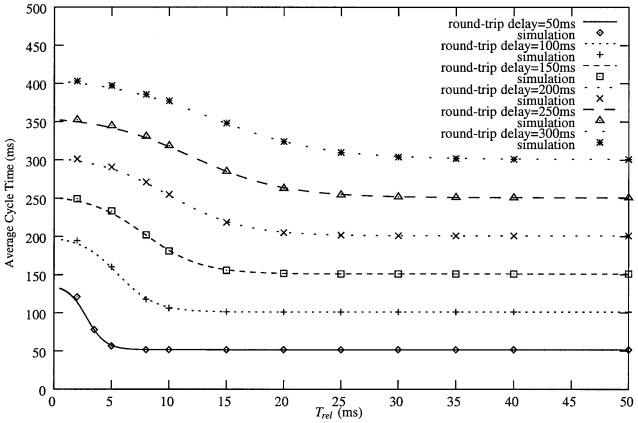


Fig. 7 Average cycle time as a function of T_{rel} under various round-trip delays.

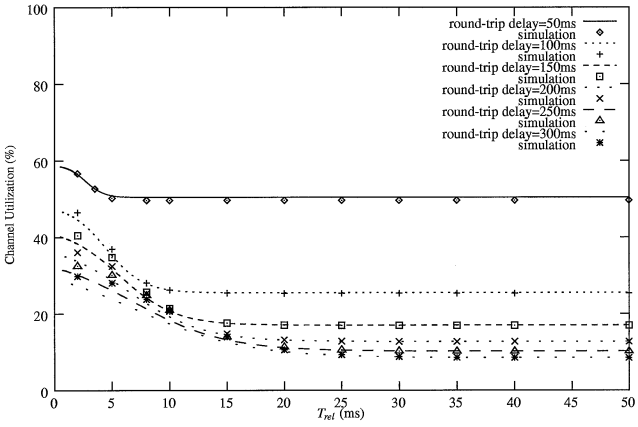


Fig. 8 Channel utilization as a function of T_{rel} under various round-trip delays.

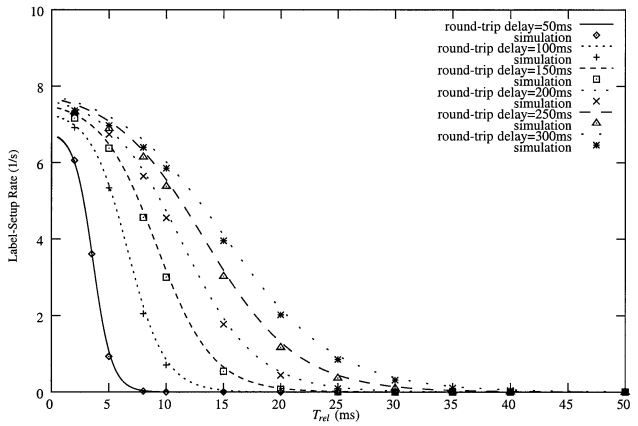


Fig. 9 Label-setup rate as a function of T_{rel} under various round-trip delays.

Fig. 8, and the corresponding throughput is in the range of 2.2 Mbps to 4.56 Mbps, as observed from Fig. 7 and Eq. (12). Now one can choose an adequate T_{rel} value according to the required throughput. The fact that channel utilization is improved with decreasing T_{rel} is

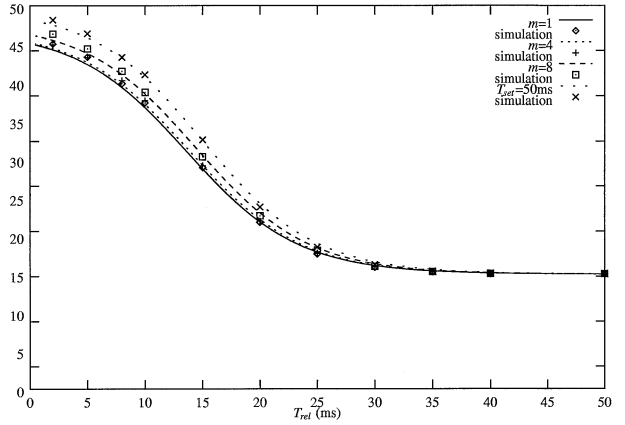


Fig. 10 Average cycle time as a function of T_{rel} with different m under 300 ms round-trip delay.

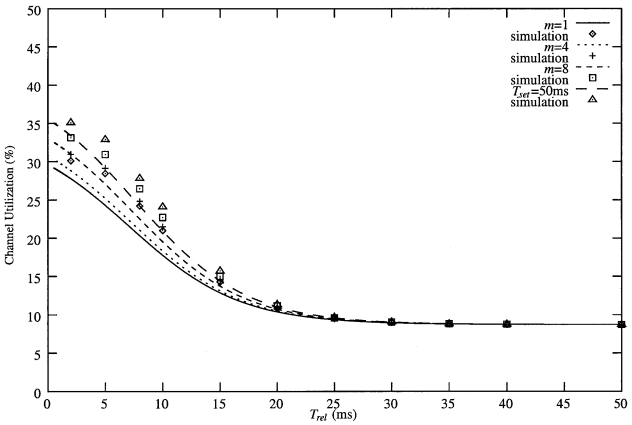


Fig. 11 Channel utilization ratio as a function of T_{rel} with different m under 300 ms round-trip delay.

due to the fast release of labels. With additional setup operations, the average cycle time is then increased and the throughput is lowered. Therefore, one need to consider the trade-off between the throughput and the channel utilization. In turn, as shown in Fig. 9, the label processing overhead can be reduced and the throughput should be improved by increasing T_{rel} because less label-setup operations are required for larger T_{rel}. However, this also results in lower channel utilization and causes more bandwidth to be wasted. This should explain why it is important to select the most appropriate T_{rel} value.

Next, we examine the behavior of the label switch under different label-setup parameters and illustrate how the system performance is improved with different policies. In Figs. 10–12, we show the average cycle time, the channel utilization, and the label-setup rate as a function of T_{rel} with different label-setup triggering threshold m. The round-trip delay 300 ms is assumed. As shown in Fig. 10, the larger the m value, the longer the average cycle time. The reason is that more packets are held in the switch buffer before the label to be

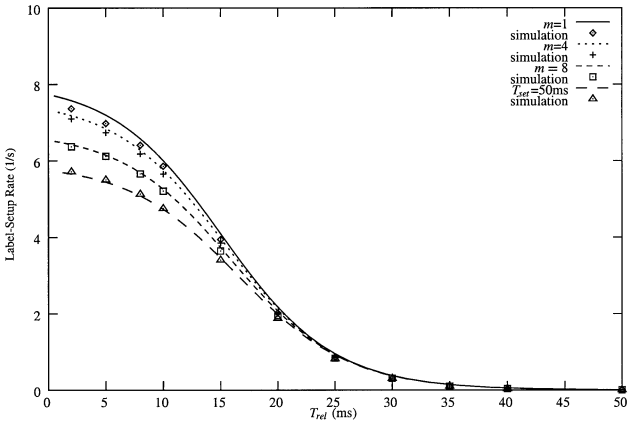


Fig. 12 Label-setup rate as a function of T_{rel} with different m under 300 ms round-trip delay.

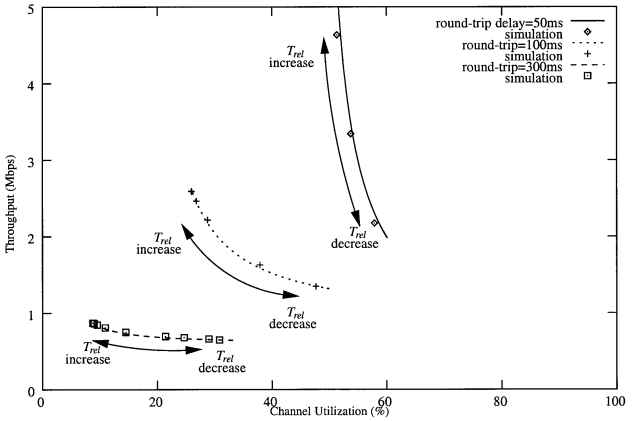


Fig. 13 Throughput verus Channel Utilization with T_{set} = 10 ms.

set up. The merit of larger m is to obtain higher channel utilization and lower label processing overhead, as shown in Figs. 11 and 12. For a larger T_{rel}, the channel utilization and the label-setup rate do not change a lot for any m. As a result, when T_{rel} is sufficiently long, m is chosen to be 1 to simplify the implementation. In turn, when T_{rel} is relatively short, the use of large m directly results in longer average cycle time and lower throughput.

Another important observation can be made from Fig. 8, with large round-trip delay (such as 300 ms), efficient label-setup and release operations can easily increase channel utilization. In turn, if the round-trip delay is small ($\leq$50 ms), the improvement of utilization is less significant. In order to illustrate the tradeoff between the throughput and the channel utilization by adjusting T_{rel} under various round-trip delays, we plot the throughput versus the channel utilization under various round-trip delays with a fixed T_{set} as shown in Fig. 13. With this figure, one not only can easily determine T_{rel} according to the required throughput and the channel utilization, but also can clearly see 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 $k = 100$. The setup timer T_{set} in the analytical model is approximated by a pseudo threshold n. This approximation technique is found to be valid according to the simulation results shown in Figs. 10–13.

6. Conclusions

The queueing model for a flow-based label switch has been developed, and the corresponding performance formula is derived. Observed from either simulations or analytical results, one can find that desiring to achieve high channel utilization and high throughput at the same time is a conflict. With our model, the tradeoff issues can be elaborated so that network engineers can reach a balance between the channel bandwidth utilization and the system throughput performance. With the proposed label-setup policy, one can achieve higher channel utilization with minimal label processing overhead by selecting appropriate timer thresholds. Alternatively, one can predict the required processing overhead under performance constraints.

If for some applications, the label-setup triggering threshold $m > 1$ results in longer delay and becomes beyond endurance, one still can focus on the selection of an appropriate label-release timer value using our proposed model with $m = 1$.

In this paper, we have only analyzed the performance of a flow-based label switch for the single IP flow with a fixed window size. Thus, this model can only reflect the contentions between various IP flows in the IP controller as they compete for limited resources, such as SVC (in IP over ATM) or labels (in MPLS) by adjusting the label-setup processing time in *Label_Setup* queue. How to adopt a more sophisticated model for resource contention, such as the vacation model, and how to illustrate the behavior of the variable-size windows of TCP should be an interesting area for further study.

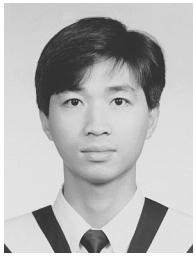
Acknowledgement

The authors would like to thank the National Science Council of the Republic of China for financially supporting this research under Grant Grant NSC 88-2213-E-002-078 and NSC 89-2213-E-002-078.

References

[1] R. Cole, D. Shur, and C. Villamizar, "IP over ATM: A framework document," RFC 1932, April 1996.
[2] Y. Rekhter, et al., "Cisco systems' tag switching architecture overview," RFC 2105, Cisco System, Inc., Feb. 1997.

[3] Y. Katsube, K. Nagami, and H. Esaki, "Toshiba's router architecture extensions for ATM: Overview," RFC 2098, Toshiba R&D Center, Feb. 1997.
[4] P. Newman, T. Lyon, and G. Minshall, "Flow labeled IP: Connectionless ATM under IP," INFOCOM'96, pp.1251–1266, San Francisco, CA.
[5] P. Newman, et al., "IP switching and gigabit routers," IEEE Commun., Mag., pp.64–69, Jan. 1997.
[6] A. Viswanathan, et al., "ARIS: Aggregate route-based IP switching," IETF Internet draft-viswanathan-aris-overview-00.txt, March 1997.
[7] E. Rosen, A. Viswanathan, and R. Callon, "Multiprotocol label switching architecture," IETF Internet draft-ietf-mpls-arch-05.txt, April 1999.
[8] R. Callon, P. Doolan, N. Feldman, A. Fredette, and G. Swallow, "A framework for multiprotocol label switching," IETF Internet draft-ietf-mpls-framework-03.txt, June 1999.
[9] P. Newman, G. Minshall, and T. Lyon, "IP switching: ATM under IP," IEEE/ACM Trans. Networking, pp.117–129, Aug. 1997.
[10] 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.
[11] 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, 1998.
[12] 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.
[13] 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.
[14] N. Endo, Z. Niu, and A. Sawada, "Performance study of SVC-based IP-over-ATM networks," IEEE ATM Workshop, pp.529–534, Kochi, Japan, 1999.
[15] J. Zheng and V.O.K. Li, "Performance model for IP switch," IEE Electron. Lett., vol.34, no.21, pp.2010–2012, Oct. 1998.
[16] S. Shakkottai, "TCP over end-to-end ABR: A study of TCP performance with end-to-end rate control and stochastic available capacity," Master of Engg. Thesis, Indian Institute of Science, Bangalore, India, Jan. 1998.
[17] P.G. Harrison and N.M. Patel, Performance Modeling of Communication Networks and Computer Architectures, pp.288–295, Addison Wesley, 1993.
[18] 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.
[19] 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.
[20] 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.



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 and ACM.