Performance Analysis of Soft–State Lightpath Management in GMPLS–Based WDM Networks

Shinya Ishida, Shin'ichi Arakawa, and Masayuki Murata
Graduate School of Information Science and Technology, Osaka University
1–5 Yamadaoka, Suita, Osaka 565–0871, Japan
Email: {s-isida, arakawa, murata}@ist.osaka-u.ac.jp
Telephone: +81–6–6879–4542, Fax: +81–6–6879–4544

Abstract

RSVP–TE is a signaling protocol to setup and teardown lightpaths in wavelength–routed GPMLS networks. RSVP–TE uses the soft–state control mechanism to manage lightpaths. In the soft–state control mechanism, each node sets a timer for each control state and resets the timer with refresh messages to maintain the state. When the timer expires due to losses of refresh messages, the control state is initialized and a reserved resource managed with the state is released. It has been considered that resource utilization of soft–state protocols is inferior to that of hard–state protocols since soft–state protocols may reserve resources until control states are deleted due to timeout. Therefore, some extensions to promote the performance of soft–state protocols, such as message retransmission, have been considered. In this paper, we analyze the behavior of GMPLS RSVP–TE and its variants with a Markov model and analyze the performance of RSVP–TE. From the results, we demonstrate that resource utilization of RSVP–TE can be equivalent to that of a hard–state protocol when the loss probability of signaling messages is low. We also investigate the effectiveness of message retransmission and show that using message retransmission leads to poor resource utilization in some cases.

1. Introduction

Lightpaths are data channels for transfering data packets or data streams in wavelength–routed networks. A lightpath is established by reserving a wavelength of each link along a route from a source node to a destination node. When a wavelength of a link is reserved, an optical switch connected to the link is configured. Each node consists of two parts: a data plane and a control plane. A data plane includes optical switches connected with optical fibers, while a control plane exchanges signaling messages in–band or out–band and configures states of optical switches, according to a signaling protocol. GMPLS (Generalized Multi–Protocol Label Switching) [1] is a protocol to manage lightpaths in wavelength–routed networks. RSVP–TE (Resource reSerVation Protocol – Traffic Engineering) [2] is a soft–state signaling protocol for GMPLS networks. In soft–state signaling, each node sets timers for control states and initializes control states when corresponding timers expire. If a node receives a refresh message before a timer expires, it resets the timer and maintains the corresponding state. Since reserved resources are released due to timeout, resource utilization would be worse than that in hard–state control. In addition, soft–state signaling requires more signaling messages than hard–state signaling in order to refresh states. However, nodes managing states in soft–state control initialize states even when signaling messages do not reach them due to network failures. In actual networks, not only message losses but also control plane failures would occur. Therefore, soft–state management is required to achieve high network availability.

Many signaling protocols for lightpath establishment in wavelength–routed networks have been proposed: BR (Backward Reservation) [9], FR (Forward Reservation) [9], IIR (Intermediate–Initiated Reservation) [6], and PR (Parallel Reservation) [7]. The main purpose of these works has been to improve blocking performance. These protocols have been evaluated as hard–state signaling protocols since it is supposed that signaling messages are never lost in those performance evaluations. In hard–state signaling, states are managed by explicit signaling messages; that is, nodes continue to reserve unnecessary wavelengths when signaling messages are lost. An infrequent lack of signaling messages could be dealt with by message retransmission. However, when nodes cannot communicate with each other due to failures of their control planes or for some other reasons, unnecessary wavelengths are not released until the control plane is recovered. Resource utilization thus deteriorates.

In [5], five types of signaling class, the pure soft–state, pure soft–state with three types of extensions, and the pure hard–state, are modeled with a Markov chain. The authors also analyze the inconsistency ratio, which is the probability that states of a source node and a destination node are not consistent, of each signaling class by using steady–state probabilities. However, their model cannot be applied to the analysis of GMPLS RSVP–TE because they consider only the forward control state, which is delivered from source nodes to destination nodes. Furthermore, the relation between the inconsistency ratio and network performance is unclear.

In this paper, we investigate how control parameter settings for GMPLS RSVP–TE affect the network performance and when the message retransmission of GMPLS RSVP–TE works effectively. To more precisely understand the influence of each control parameter to the network performance and the relation between control parameter settings, we extend the Markov model in [5] for GMPLS RSVP–TE. Using the Markov model, we can describe the behavior of GMPLS RSVP–TE in detail and can analyze the steady–state probabilities of an LSP (Label Switched Path) session. We then investigate the network performance, such as resource utilization and LSP setup delay of GMPLS RSVP–TE. From our numerical analyses, we demonstrate that the resource utilization of RSVP–TE can be equivalent to those of hard–state protocols when the loss probability of signaling messages is relatively low. We also examine the effectiveness of message retransmission and reveal that the use of such message retransmission can result in poor resource utilization in some cases.

This paper is organized as follows. We describe a brief explanation of RSVP–TE in Section 2. In Section 3, we develop an RSVP–TE model for a single–hop LSP and analyze the performance of the standard RSVP–TE, an extended RSVP–TE with the message retransmission, and the hard–state–based backward reservation. Section 4 extends the model for a multi–hop LSP, and in Section 5 we investigate the effectiveness of message retransmission for RSVP–TE. We summarize this paper in Section 6.

2. GMPLS RSVP–TE

GMPLS is the standard technology to configure lightpaths in wavelength–routed networks. In GMPLS, wavelengths are regarded as labels and lightpaths are called LSPs (Label Switched Paths). RSVP–TE is a signaling protocol for managing LSPs. In this section, we briefly review RSVP–TE.

Table 1. Types of RSVP–TE control messages

Type	Role of message
Path	request for a LSP session
Resv	reserves a label
PathErr	notifies an error relating to Path state
ResvErr	notifies an error relating to Resv state
PathTear	removes a Path state
ResvTear	removes a Resv state
ResvConf	confirms the LSP establishment

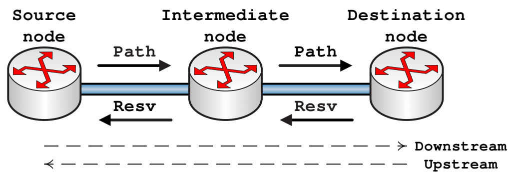


Figure 1. LSP establishment by RSVP–TE

2.1. Signaling Process of GMPLS RSVP–TE

RSVP–TE has seven types of signaling messages: Path, Resv, PathErr, ResvErr, PathTear, ResvTear, and ResvConf as listed in Table 1. Figure 1 illustrates LSP establishment by RSVP–TE, where each control signal is sent hop–by–hop. When an LSP request arrives at a source node, the source node creates a Path trigger message and sends it downstream. Each intermediate node that receives the Path trigger message makes a Path state in itself and it also checks information about available labels in the Path trigger message. If there is an available label on the outgoing link, the node forwards the message downstream. Otherwise, a PathErr message is created and sent back toward the source node. When the Path trigger message arrives at a destination node, the node makes a Path state. If there is one or more available labels, the destination node selects a label from available labels listed in the received Path trigger message and reserves the label. Then, a Resv trigger message that includes the selected label is created and sent upstream. If there is no available label, the destination node sends a PathErr message upstream. Each intermediate node that receives the Resv trigger message reserves the label specified in the message and makes a Resv state. After that, the node selects a label to be reserved by its upstream node (if the wavelength selection is subject to the wavelength continuity constraint, the same label is selected) and forwards the Resv trigger message upstream. If an intermediate node fails to reserve a label due to a lack of available labels, the node creates a ResvErr message and sends it downstream. If the source node successfully receives the Resv trigger message, it means that an LSP is established. If the destination node requests confirmation of LSP establishment, the source node sends a ResvConf message toward the destination node. After data transmission is completed, the source node sends a PathTear message downstream. Intermediate nodes that receive the PathTear message delete their Path and Resv states and forward the message downstream.

2.2. State Control at nodes

As mentioned above, nodes create a Path and a Resv state for each LSP. In soft–state control, these states are maintained by refreshing them during data transmission. Furthermore, when nodes create control states, they also set state timeout timers to manage lifetimes of control states. If a state timeout timer expires, a corresponding control state is removed and a reserved label is released. Lifetimes of control states are prolonged and state timeout timers are reset if refresh messages arrive before state timeouts. When a node sends a Path or a Resv trigger message, it also sets a refresh timer, and every time a refresh timer expires, a refresh message is sent and the timer is reset. In GMPLS RSVP–TE, signaling messages are sent in best–effort unless the message retransmission extension [3] is used. Therefore, even if a previous refresh message is lost, a next refresh message would be sent.

Lifetimes of states are typically longer than refresh intervals so as to send some refresh messages by state timeouts. On the other hand, since hard–state signaling does not have the refresh mechanism, message retransmission is essential for delivering signaling messages to receiver nodes. Loss of a PathTear message in the standard RSVP–TE requires so much as a state lifetime in order to release a reserved label. Therefore, RSVP–TE would make the resource utilization lower than by hard–state signaling. Although short lifetimes of control states may improve the resource utilization of RSVP–TE, refresh intervals also become short at the same time, which increases the quantity of signaling messages. If several losses of refresh messages occur, corresponding control states are removed incorrectly (false removal). Although frequent refreshing suppresses false removals, the number of signaling messages also increases. However, RSVP–TE is tolerant to failures on the control plane. Control states would, therefore, be initialized by state timeout while control channels are down due to network failures. Hard–state signaling cannot update or delete control states during such failures on the control plane.

3. Modeling and Analysis of GMPLS RSVP–TE for Single–Hop LSP

In this section, we investigate the steady–state performance of GMPLS RSVP–TE for single–hop LSP. We develop a model of GMPLS RSVP–TE based on the Markov model in [5] and use it to analyze the performance of GMPLS RSVP–TE. We consider two types of RSVP–TE: the standard RSVP–TE (we call this RSVP–TE hereafter) and RSVP–TE with the extension of the message retransmission (RSVP–TE/Ack). As opposed to the model in [5], our model incorporates RSVP–TE that has the control state for backward direction, i.e., Resv state. We also extend the state transition of the control plane failure and recovery into the model to show how GMPLS RSVP–TE is stable during disruption of the communications on the control plane.

3.1. Model of GMPLS RSVP–TE for Single–Hop LSP

First, we consider the model of GMPLS RSVP–TE without control plane failure. We assume the following in order to develop our models with the Markov chain.

- Arrivals of LSP setup requests follow a Poisson process with rate λ_r.
- Connection time of LSPs follows an exponential distribution with rate μ.
- Message processing delay at nodes is 0.
- Propagation delay per hop of signaling messages follows an exponential distribution with rate $1/D$.
- Signaling message loss probability per hop p_l and blocking probability of label reservation per hop p_b are constant.
- Any incoming wavelength can be converted to any outgoing wavelength.

We also assume the items below for the control parameters and the message processing of RSVP–TE.

- Refresh intervals follow an exponential distribution with rate $1/T$ regardless of sender nodes and message types.
- Lifetimes of control states X are given as T multiplied by k, i.e., $X = kT$, where k is a constant number of refresh events.
- Retransmission intervals follow an exponential distribution with rate $1/R$ regardless of the sender node and message type.
- The maximum number of retransmission times m is constant.
- Error messages are not lost.
- Acknowledgments of message receipt are not lost.

Now we focus on the steady–state behavior of GMPLS RSVP–TE. Although we assume that the time parameters, propagation delay, refresh interval, state lifetime, and retransmission interval follow exponential distributions, the

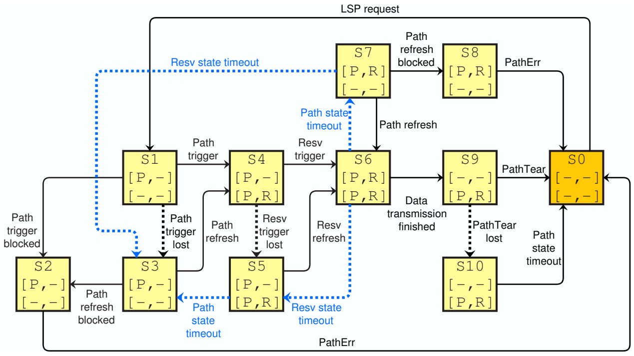


Figure 2. State transition of RSVP–TE for a single–hop LSP

average performance of GMPLS RSVP–TE is decided from the average values of those parameters, i.e., D, T, X, and R. Hence, these assumptions do not affect to the results we want. Constant message loss probability and blocking probability are also reasonable for the same reason that we are paying attention to the steady state. The time unit is seconds.

Figure 2 shows the state transition of RSVP–TE for a single–hop LSP. This state transition consists of 11 states: S_i $(i = 0, 1, \cdots, 10)$. Each square represents a state of the state transition and has a 2×2 matrix. The first row of the matrix has the status of a source node, and the second row has the status of a destination node. A "P" in the left column of a state indicates that there is a Path state. Similarly, "R" in the right column indicates that there is a Resv state. If there is no control state (i.e., a default state), it is indicated as "$-$." We explain the operations of RSVP–TE at S_i below.

S_0: The initial state. When an LSP setup request arrives at a source node, the Markov chain transits to S_1.

S_1: The source node creates a Path state and sends a Path trigger message. If the message is lost on the way from the source node to a destination node, the Markov chain transits to S_3. If the destination node successfully receives the message and if there is an available label, the Markov chain transits to S_4. If a destination node receives the message but there is no available label, the Markov chain transits to S_2.

S_2: The destination node sends a PathErr message. The Markov chain transits to S_0.

S_3: The source node sends a Path refresh message. If the destination node receives the message and there is an available label, the Markov chain transits to S_4. If the destination node receives the message and there is no available label, the Markov chain transits to S_2.

Table 2. Transition rates of the state transition

Transition	Rate (RSVP–TE)	Rate (RSVP–TE/Ack)
$S_0 \to S_1$	λ_r	
$S_1 \to S_2$	$\frac{p_b(1-p_l)}{D}$	
$S_1 \to S_3$, $S_2 \to S_3$, $S_4 \to S_5$, $S_9 \to S_{10}$	$\frac{p_l}{D}$	
$S_1 \to S_4$	$\frac{(1-p_b)(1-p_l)}{D}$	
$S_2 \to S_0$, $S_4 \to S_6$, $S_8 \to S_0$, $S_9 \to S_0$	$\frac{1-p_l}{D}$	
$S_3 \to S_2$, $S_7 \to S_8$	$\frac{p_b(1-p_l)}{T}$	$p_b(1-p_l)(\frac{1}{T} + \frac{1}{R})$
$S_3 \to S_4$, $S_7 \to S_6$	$\frac{(1-p_b)(1-p_l)}{T}$	$(1-p_b)(1-p_l)(\frac{1}{T} + \frac{1}{R})$
$S_6 \to S_9$	μ	
$S_5 \to S_3$, $S_6 \to S_5$, $S_6 \to S_7$, $S_7 \to S_3$	$\frac{p_l^k}{X}$	$\frac{p_l^{(k-1)(m+1)+1}}{X}$
$S_{10} \to S_0$	$\frac{1}{X}$	$\frac{1-p_l}{R} + \frac{1}{X}$

S_4: The destination node creates a Path state. The destination node also makes a Resv state and sends a Resv trigger message. If the source node receives the Resv trigger message, the Markov chain transits to S_6. Otherwise, the Markov chain transits to S_5.

S_5: The destination node sends a Resv refresh message. If the source node receives the Resv refresh message, the Markov chain transits to S_6. If a false removal occurs at the destination node because of the successive loss of refresh messages, the Markov chain transits to S_3.

S_6: In this state, the source node is transmitting data by the established LSP. If the data transmission is successfully completed, the Markov chain transits to S_9. If a false removal of either the Resv state at the source node or the Path state at the destination occurs, the Markov chain transits to S_5 or S_7, respectively.

S_7: If the destination node receives a Path refresh message and there is an available label, the Markov chain transits to S_6. If the destination node receives a Path refresh message and there is no available label, the Markov chain transits to S_8. If a false removal occurs at the source node, the Markov chain transits to S_3.

S_8: The destination node sends a PathErr message. The Markov chain transits to S_0.

S_9: The source node sends a PathTear message. If the destination node receives the message, the Markov chain transits to S_0. Otherwise, the Markov chain transits to S_{10}.

S_{10}: If a Path state at the destination node is deleted by a state timeout, the Markov chain transits to S_0.

The transition rates of the state transition in Fig. 2 are listed in Table 2. $S_{i_1} \to S_{i_2}$ represents the transition from

S_{i_1} to S_{i_2}.

The state transition of RSVP–TE/Ack is obtained by some replacements of the transition rates of RSVP–TE as in Table 2. The retransmission rate in RSVP–TE/Ack is given as $1/R$; therefore, the rate that refresh messages are sent in RSVP–TE/Ack is $1/T + 1/R$. RSVP–TE/Ack can also retransmit teardown messages. The rate of $S_{10} \rightarrow S_0$ in RSVP–TE/Ack is $1/X + (1 - p_l)/R$ since the probability that a retransmitted message reaches the receiver node is $(1 - p_l)$.

The hard–state BR does not use timers or refresh messages; and the rate that signaling messages are retransmitted in the hard–state BR is $1/R$. The state transition of the hard–state BR is obtained by replacing the transition rates of RSVP–TE/Ack, that is, replacing $1/T$ and $1/X$ with 0. Then, states S_7 and S_8 become unreachable and can be removed.

3.2. Model of GMPLS RSVP–TE for Single–Hop LSP with Control Plane Failure

Here we consider the model of GMPLS RSVP–TE with control plane failure. To develop this model, we add the following assumptions.

- When a failure occurs on a control plane, all the communications of signaling messages among the nodes become impossible. This is the worst case of control plane failure.
- When a source node finds that a failure has occurred in a control plane, the source node deletes its Path state immediately.
- In our analysis, we set control plane failures to occur in accordance with a Poisson process with rate ϕ, and the delays to recover from control plane failures follow an exponential distribution with rate γ.

Figure 3 shows the state transition of RSVP–TE for a single–hop LSP with control plane failure. Two new states, S_{11} and S_{12}, and their associated transitions are added to the state transition in Fig. 2. Control plane failures would occur at S_3, S_5, S_6, and S_{10}. At S_3, if a control plane failure occurs, the Markov chain transits to S_{12}. While at the other states, if a control plane failure occurs, the Markov chain transits to S_{11}. RSVP–TE works at S_{11} and S_{12} as follows.

S_{11}: If a control plane recovers from a failure, the Markov chain transits to S_{10}. If the Path state at the destination node is deleted by a state timeout, the Markov chain transits to S_{12}.

S_{12}: If a control plane recovers from a failure, the Markov chain transits to S_0.

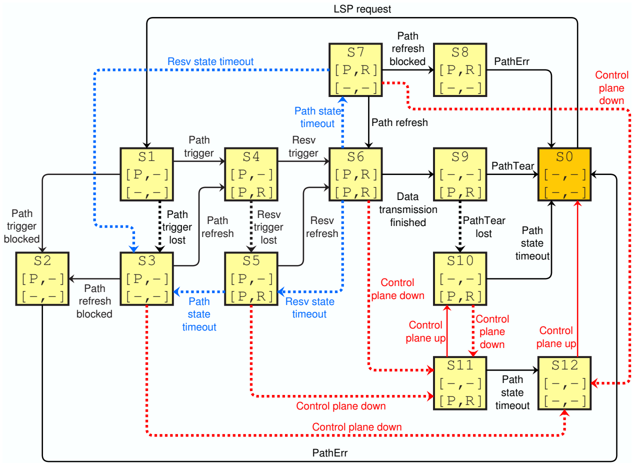


Figure 3. State transition of GMPLS RSVP–TE for a single–hop LSP with control plane failure

Table 3. Rates of the additional transitions for control plane failure

Transition	Rate: RSVP–TE	Rate: RSVP–TE/Ack
$S_3 \rightarrow S_{12}$, $S_5 \rightarrow S_{11}$, $S_6 \rightarrow S_{11}$, $S_7 \rightarrow S_{12}$, $S_{10} \rightarrow S_{11}$	ϕ	
$S_{11} \rightarrow S_{10}$, $S_{12} \rightarrow S_0$	γ	
$S_{11} \rightarrow S_{12}$	$\frac{1}{X}$	

The rates of the added transitions are listed in Table 3. The state transitions of RSVP–TE/Ack and the hard–state BR are obtained in the same way as in Sec. 3.1.

3.3. Analysis of GMPLS RSVP–TE for Single–Hop LSP

We analyze the performance of GMPLS RSVP–TE with our models given in Sec. 3.1 and 3.2. Unoccupied time is used as the performance index for this analysis. Unoccupied time is the time that a label is reserved but not used for data transmission. The longer the unoccupied time is, the lower the resource utilization becomes. This unoccupied time is obtained by using the steady–state probabilities. Supposing that the state transition of GMPLS RSVP–TE is composed of N states, π_i is the steady–state probability for S_i $(i = 0, 1, \cdots, N-1)$, and t_i is the average total time that the process of GMPLS RSVP–TE is at S_i. Let T be the average duration from the beginning to the end of GMPLS RSVP–TE sessions. A GMPLS RSVP–TE session starts when a source node sends a Path trigger message to establish an

LSP and finishes when the LSP is removed after the data transmission. Here, t_i is expressed as

$$t_i = \pi_i T.$$

From this equation, the relation between any two steady–state probabilities can be described as

$$\frac{\pi_i}{\pi_d} = \frac{t_i}{t_d} \quad (i, j = 0, 1, \cdots, N-1).$$

Since the average time of data transmission is $1/\mu$,

$$t_i = \frac{\pi_i}{\mu \pi_d},$$

where S_d is the state that a source node transmits data on an established LSP. The steady–state probabilities can be obtained by solving the state transition equation. Let $S^{'}$ be a set of the states for which a label is reserved but unoccupied for data transmission. The unoccupied time $T^{'}$ is defined as follows:

$$T^{'} = \sum_{i \in I^{'}} t_i = \sum_{i \in I^{'}} \frac{\pi_i}{\mu \pi_d} \quad (I^{'} = \{i \mid S_i \in S^{'}\}).$$

In the state transition in Fig. 2, the states having a Resv state are $S_4, S_5, \cdots, S_{10}$. Since the state that a source node transmits data to the destination node is S_6, $T^{'}$ is,

$$T^{'} = \frac{\pi_4 + \pi_5 + \pi_7 + \pi_8 + \pi_9 + \pi_{10}}{\mu \pi_6}. \quad (1)$$

For the state transition of Fig. 3, $T^{'}$ is given by

$$T^{'} = \frac{\pi_4 + \pi_5 + \pi_7 + \pi_8 + \pi_9 + \pi_{10} + \pi_{11}}{\mu \pi_6}. \quad (2)$$

The arrival rate of LSP requests has no impact on the unoccupied time since T is the average duration from the beginning to the end of the GMPLS RSVP–TE sessions. Hence, we merged S_0 and S_1 into a state and solved the state transition equation. We compare the unoccupied times of five signaling protocols in Table 4. RSVP–TE(A) is a variant of RSVP–TE, whose refresh interval is as short as the retransmission interval of RSVP–TE/Ack. Note that the state lifetime of RSVP–TE(A) is also shortened to 1.5 sec from 90 sec. RSVP–TE(B) has the same refresh interval as RSVP–TE(A) and the same state lifetime as RSVP–TE. HS–BR is BR with hard–state control that has the same retransmission interval as RSVP–TE/Ack. Since the message retransmission continues until a sender node confirms that the signaling message has been received by the receiver node in HS–BR, the maximum number of retransmission times is unlimited. In what follows, we use these parameter values unless otherwise specified: $D = 0.001$, $T = 30$, $k = 3$, $\mu = 0.00001$, $p_l = 0.00001$, $p_b = 0.001$, $R = 0.5$,

Table 4. Definitions of protocols and their parameter settings

Protocol	T	k	R	m
RSVP–TE	30	3	–	–
RSVP–TE(A)	0.5	3	–	–
RSVP–TE(B)	0.5	180	–	–
RSVP–TE/Ack	30	3	0.5	3
HS–BR	–	–	0.5	∞

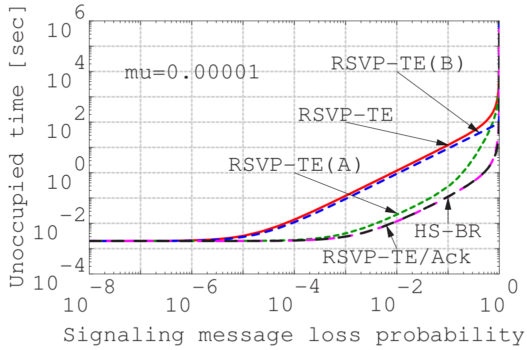


Figure 4. Unoccupied time versus message loss probability for a single–hop LSP without control plane failure

and $m = 3$. D does not affect the increase of LSP setup and teardown delays but just decides the minimum of those delays. The default values of T, k, R, and m are described as standard or reference values in [3, 4].

There are three factors that control whether reserved labels remain unoccupied in RSVP–TE: propagation delay, signaling message loss, and false removal. Propagation delay, D, is unavoidable and thus determines the minimum unoccupied time. Signaling message loss occurs with the probability p_l. If p_l is not small enough, the unoccupied time is increased by signaling message loss. The probability that a false removal occurs is proportional to the message loss probability to the power of n, p_l^n ($n = k$ for RSVP–TE; $n = (k-1)(m+1) + 1$ for RSVP–TE/Ack). Meanwhile, the unoccupied time of HS–BR has nothing to do with false removal because HS–BR does not use any timers.

Figure 4 shows the unoccupied time, which is dependent on the signaling message loss probability for a single–hop LSP without control plane failure. When the signaling message loss probability is smaller than 10^{-6}, there is no difference in the unoccupied time among the five protocols since message losses seldom occur. When the message loss probability is greater than 10^{-6}, the increase of unoccupied time in RSVP–TE is mainly due to losses of PathTear messages. In RSVP–TE, since PathTear messages are not retransmitted, if a PathTear message is lost, control states at a destination node are not deleted until the state timeout timer expires. RSVP–TE(A) and RSVP–TE(B) do not retransmit signaling messages, though the performance degradation of RSVP–TE(A) is less than those of RSVP–TE and RSVP–TE(B) since the state lifetime of RSVP–TE(A) is quite short. The difference in unoccupied time between RSVP–TE and RSVP–TE(B) comes from occurrences of false removals. False removals are likely to occur when the message loss probability is high. According to Fig. 4, the influence of false removal does not appear if the message loss probability is lower than 0.1.

The results of RSVP–TE/Ack exhibit a similar tendency as HS–BR, where the unoccupied time of RSVP–TE/Ack is shorter than that of RSVP–TE(A) since RSVP–TE/Ack can retransmit PathTear messages. In addition, the retransmission of refresh messages enables RSVP–TE/Ack to avoid false removals even when the message loss probability is high.

At this point we investigate the performance of GMPLS RSVP–TE for a single–hop LSP with control plane failure. We analyzed the unoccupied time in these four cases (1 day $= 86,400$ sec $< 10^5$ sec. 3 year $= 93,312,000$ sec $< 10^8$ sec).

- Case 1: Control plane failures rarely occur and it does not take a long time for the control plane to recover from a failure ($\phi = 10^{-8}$ and $\gamma = 10^{-2}$. The annual operating ratio of the control plane is 99.9999%).
- Case 2: Control plane failures rarely occur and it takes a long time for the control plane to recover from a failure ($\phi = 10^{-8}$ and $\gamma = 10^{-5}$. The annual operating ratio of the control plane is 99.9%).
- Case 3: Control plane failures frequently occur and it does not take a long time for the control plane to recover from a failure ($\phi = 10^{-5}$ and $\gamma = 10^{-2}$. The annual operating ratio of the control plane is 99.9%).
- Case 4: Control plane failures frequently occur and it takes a longer time for the control plane to recover from a failure than in case 3 ($\phi = 10^{-5}$ and $\gamma = 10^{-3}$. The annual operating ratio of the control plane is 99%).

Figure 5 shows the unoccupied times in these four cases. As can be seen from the comparison between Fig. 4 and Fig. 5(a), the influence of control plane failure does not appear in Case 1. However, Fig. 5(b) shows that the performance of HS–BR decreases even when the message loss probability is low. This is because HS–BR does not have the state timeout mechanism and must wait until the control plane recovers in order to release the reserved resources. This tendency can also be seen in Case 3 (Fig. 5(c)) and Case 4 (Fig. 5(d)), where control plane failures occur frequently. On the other hand, the unoccupied time of RSVP–TE is independent of the recovery time. The unoccupied times of RSVP–TE in Cases 1 and 2 are almost the same, and there is no difference between the unoccupied times of RSVP–TE in Cases 3 and 4, too. These results indicate that the soft–state protocols are stable in terms of control plane failures.

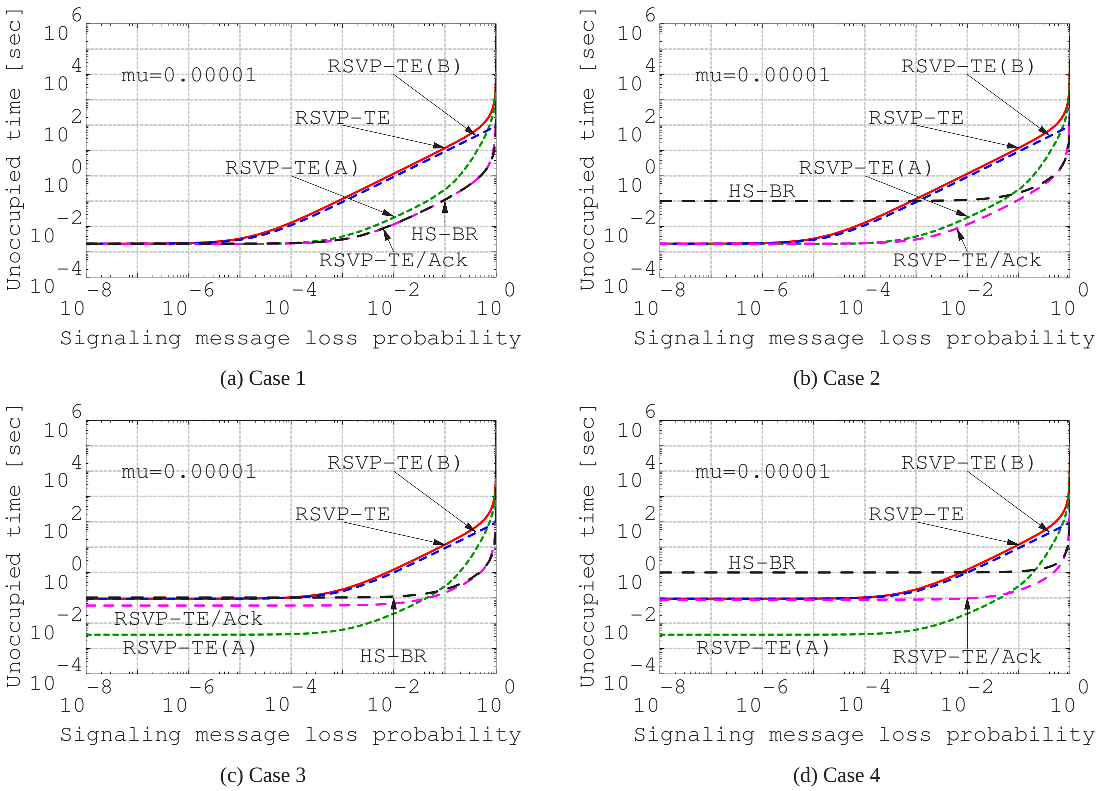
4. Model and Analysis of GMPLS RSVP–TE for Multi–Hop LSP

In this section, we develop the model of GMPLS RSVP–TE for multi–hop LSPs and analyze LSP setup delay, recovery delay, and teardown delay. LSP setup delay is the time from when a source node sends a Path trigger message till when an LSP is established. Recovery delay is the time from when an LSP is disrupted by a false removal till when the disrupted LSP recovers. Teardown delay is the time from when a source node sends a PathTear message till when an LSP is completely deleted. We do not discuss the control plane failure here but it can be extended to our model, as in Sec. 3.2.

4.1. Model of GMPLS RSVP–TE for Multi–Hop LSP

To analyze the performance of GMPLS RSVP-TE for multi–hop LSPs, we assume that false removals never occur during the LSP setup and recovery phase. That is, we consider false removals only when the LSP is established. Although we can develop the Markov model without this assumption, the number of states rapidly increases with an increasing number of hops. This is because states have to be prepared based on where and when false removals occur. Furthermore, since the LSP holding time (an order of seconds or more) is longer than the LSP setup delay (in the order of ms), the impact of false removals during the LSP setup phase would be small. Actually, the probability that a false removal occurs is quite low in the single–hop case (see the difference between RSVP–TE and RSVP–TE(B) in Fig. 4). Therefore, we assume here that false removals occur after a LSP is successfuly established. To enable our model to analyze the recovery time, we also assume that a disrupted LSP is recovered on the same route after a false removal occurs.

Figure 6 illustrates the state transition of RSVP–TE for an h–hop LSP, where rectangles represent the states and the number of states is $14h$. The index of state S_i, i, is denoted inside each rectangle. The process of setting up an LSP setup is modeled with the states S_1 to S_{6h-1}, while the process of recovery from a false removal is modeled with the states S_{6h+1} to S_{12h-1}, and LSP teardown is modeled



(a) Case 1 (b) Case 2 (c) Case 3 (d) Case 4

Figure 5. Unoccupied time versus message loss probability for a single–hop LSP with control plane failure

with the states S_{12h} to S_{14h-1}. We explain the operations of RSVP–TE at each state below, skipping the explanations of states S_{6h+1} to S_{12h-1} since the transitions among these states are same as the transitions among the states S_1 to S_{6h-1}.

S_0: The initial state. When an LSP setup request arrives at a source node, the Markov chain goes to S_1.

S_1: The source node makes a Path state and sends a Path trigger message downstream. If the message is lost, the Markov chain goes to S_2. If a downstream node receives the message and there is an available label, the Markov chain goes to S_3. If a downstream node receives the message but there is no available label, the Markov chain goes to S_5.

S_2: The source node sends a Path refresh message. If a downstream node receives the message and there is an available label, the Markov chain goes to S_3. If the downstream node receives the message but there is no available label, the Markov chain goes to S_5.

S_{3j}: Each intermediate node makes a Path state and sends a Path trigger message. If the downstream node receives the message and there is an available label, the Markov chain goes to S_{3j+3}. If the downstream node receives the message and there is no available label, the Markov chain goes to S_{3j+5}. If the message is lost, the Markov chain goes to S_{3j+1}. $j = 1, 2, \cdots, h-1$.

S_{3j+1}: Each intermediate node sends a Path refresh message. If a downstream node receives the message and there is an available label, the Markov chain goes to S_{3j+3}. If a downstream node receives the message and there is no available label, the Markov chain goes to S_{3j+5}. $j = 1, 2, \cdots, h-1$.

S_{3j+2}: Each intermediate node sends a PathErr message. the Markov chain goes to S_{3j-1}. $j = 1, 2, \cdots, h-1$.

S_{3h}: A destination node creates a Path state. The destination node also creates a Resv state and sends a Resv trigger message. If an upstream node receives the message and reserves a label, the Markov chain goes to S_{3h+3}. If an upstream node fails to reserve a label, the Markov chain goes to S_{3h+5}. If the message is lost, the Markov chain goes to S_{3h+1}.

S_{3h+1}: The destination node sends a Resv refresh message. If an upstream node receives the message and reserves a label, the Markov chain goes to S_{3h+3}. If an upstream node fails to reserve a label, the Markov chain goes to S_{3h+5}.

S_{3h+2}: The destination node sends a PathErr message. The Markov chain goes to S_{3h-1}.

S_{3h+3j}: Each intermediate node sends a Resv trigger message. If an upstream node receives the message and reserves a label, the Markov chain goes to $S_{3h+3j+3}$. If an upstream node fails to reserve a label, the Markov chain goes to $S_{3h+3j+5}$. If the message is lost, the Markov chain goes to $S_{3h+3j+1}$. $j = 1, 2, \cdots, h-2$.

$S_{3h+3j+1}$: Each intermediate node sends a Resv refresh message. If an upstream node receives the message and reserves a label, the Markov chain goes to $S_{3h+3j+3}$. If an upstream node fails to reserve a label, the Markov chain goes to $S_{3h+3j+5}$. $j = 1, 2, \cdots, h-2$.

$S_{3h+3j+2}$: Each intermediate node sends a ResvErr message downstream. The Markov chain goes to $S_{3h+3j-1}$. $j = 1, 2, \cdots, h-1$.

S_{6h-3}: An intermediate node sends a Resv trigger message to the source node. If the source node receives the message, the Markov chain goes to S_{6h}. Otherwise, the Markov chain goes to S_{6h-2}.

S_{6h-2}: An intermediate node sends a Resv refresh message to the source node. If the source node receives the message, the Markov chain goes to S_{6h}.

S_{6h}: An LSP is established in this state. If the data transmission is completed, the Markov chain goes to S_{12h}. If a Path state at the first node from the source node is deleted by false removal, the Markov chain goes to S_{6h+2}. If a Path state at the i–th node from the source node is deleted by false removal, the Markov chain goes to $S_{6h+3j-2}$ ($j = 2, 3, \cdots, h$). If a Resv state at the i–th node from the destination node is deleted by false removal, the Markov chain goes to $S_{9h+3j-2}$ ($j = 1, 2, \cdots, h$).

S_{12h}: The source node sends a PathTear message. If a downstream node receives the message, the Markov chain goes to S_{12h+2}. If the message is lost, the Markov chain goes to S_{12h+1}.

S_{12h+1}: A Path state at the node next to a source node is deleted by state timeout. The Markov chain goes to S_{12h+2}.

S_{12h+2j}: Each intermediate node sends a PathTear message. If a downstream node receives the message, the Markov chain goes to $S_{12h+2j+2}$. If the message is lost, the Markov chain goes to $S_{12h+2j+1}$. $j = 1, 2, \cdots, h-2$.

$S_{12h+2j+1}$: A Path state at a i–th node is deleted by state timeout. The Markov chain goes to $S_{12h+2j+2}$. $j = 1, 2, \cdots, h-2$.

S_{14h-2}: A Path state at the penultimate node sends a PathTear message. If the destination node receives the message, the Markov chain goes to S_0. If the message is lost, the Markov chain goes to S_{14h-1}.

S_{14h-1}: A Path state at the destination node is deleted by state timeout. The Markov chain goes to S_0.

4.2. Analysis of GMPLS RSVP–TE for Multi–Hop LSP

We can analyze the setup delay, the recovery delay, and the teardown delay for an LSP, T_S, T_R, and T_D, by the model described above. As we discussed in Sec. 3, these delays are obtained with fractions of the steady–state probabilities:

$$T_S = \frac{\sum_{j=1}^{6h-1} \pi_j}{\mu \pi_{6h}}, \quad T_R = \frac{\sum_{j=6h+1}^{12h-1} \pi_j}{\mu \pi_{6h}}, \quad T_D = \frac{\sum_{j=12h}^{14h-1} \pi_j}{\mu \pi_{6h}}. \quad (3)$$

Figure 7 compares the LSP setup delay between a single–hop LSP and a 20–hop LSP. The horizontal axes represent the loss probability of signaling messages, and the vertical axes represent the LSP setup delay. Although setup delays are different due to the propagation delay, the points at which the setup delays of RSVP–TE and RSVP–TE/Ack start to rise are almost the same (10^{-6} for RSVP–TE and 10^{-4} for RSVP–TE/Ack). That is, the properties of RSVP–TE and RSVP–TE/Ack in regard to the signaling message loss probability are independent of LSP length. This means that the results of our analysis in Sec. 3 are applicable for discussing the effectiveness of RSVP–TE and RSVP–TE/Ack for multi–hop LSPs. Although we confirmed this observation for LSP recovery delay and teardown delay, these results are not presented here due to the page limitation.

5. Effectiveness of Message Retransmission

In previous sections, we compared RSVP–TE with RSVP–TE/Ack in instances where the signaling message

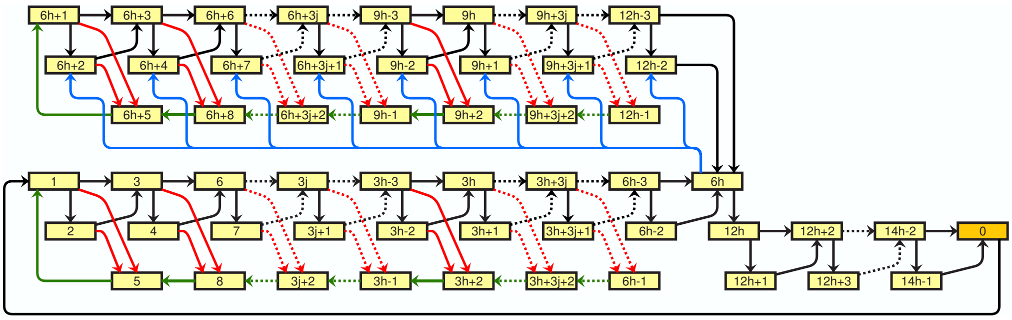


Figure 6. State transition of RSVP–TE for an h–hop LSP

loss probabilities are the same. However, the quantity of signaling messages in RSVP–TE/Ack is greater than that in RSVP–TE since signaling messages would be retransmitted in RSVP–TE/Ack. Since the size of the receive buffer is finite, if the quantity of signaling messages increases, the signaling message loss probability also increases. In this section, we reconsider the effectiveness of message retransmission in RSVP–TE/Ack taking into account the increment of message loss probability by message retransmission. We apply the results of our analysis in Sec. 3 to show when message retransmission is efficient and when it is inefficient.

5.1. Modeling of Signaling Message Loss

It is assumed that losses of signaling messages occur only due to the buffer overflow in the receive buffer. We also assume that the signaling messages in RSVP–TE arrive according to the Poisson process with rate λ_1 and that the processing time of a signaling message follows the exponential distribution with rate μ_p. When there are w LSP sessions, the total message transmission rate is $w\lambda_1$. Therefore, the message loss probability of RSVP–TE, P_{b_1}, is described with the $M/M/1/K$ queuing model:

$$P_{b_1} = \frac{(w\rho_1)^K}{\sum_{i=0}^{K}(w\rho_1)^i} = \frac{(1-w\rho_1)(w\rho_1)^K}{1-(w\rho_1)^{K+1}}, \tag{4}$$

where ρ_1 is defined as λ_1/μ_p. For RSVP–TE/Ack, the message loss probability, P_{b_2}, is given in the same manner. That is:

$$P_{b_2} = \frac{(w\rho_2)^K}{\sum_{i=0}^{K}(w\rho_2)^i} = \frac{(1-w\rho_2)(w\rho_2)^K}{1-(w\rho_2)^{K+1}}, \tag{5}$$

where $\rho_2 = \lambda_2/\mu_p$, and λ_2 is the arrival rate of signaling messages in RSVP–TE/Ack. Solving Eq. (4) for K,

$$K = \frac{\log\left[\dfrac{P_{b_1}}{1-(1-P_{b_1})w\rho_1}\right]}{\log\left[w\rho_1\right]} \tag{6}$$

is obtained. Then, P_{b_2} is expressed as a function of P_{b_1} by substituting Eq. (6) into Eq. (5).

In RSVP–TE protocols, signaling messages are sent in the forward (from a source node to a destination) and backward directions. Here we focus only on the signaling messages sent in the forward direction. In the state transition of Fig. 2, Path and PathTear fall into such messages. Path trigger messages are sent at state S_1 in Fig. 2 at a rate of $1/D$, while Path refresh messages are sent at states S_3, S_5, S_6, and S_7. PathTear messages are sent at state S_9. Hence, λ_1 is given as

$$\lambda_1 = \frac{1}{D}(\pi_1+\pi_9) + \frac{1}{T}(\pi_3+\pi_5+\pi_6+\pi_7).$$

In RSVP–TE/Ack, Path messages would be retransmitted at the rate of $1/R$ at states S_3, S_5, S_6, and S_7, and PathTear messages would also be retransmitted at $1/R$ at state S_{10}. Thus, λ_2 is given as

$$\lambda_2 = \lambda_1 + \frac{1}{R}(\pi_3+\pi_5+\pi_6+\pi_7+\pi_{10}).$$

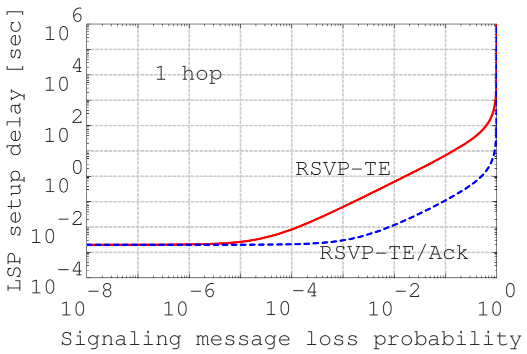
5.2. Numerical Examples

The average connection time of LSP is 100,000 sec since $\mu = 0.00001$. This is sufficiently large that $\pi_i/\pi_6 \approx 0$ $(i = 1, 2, \cdots, 10, i \neq 6)$. Therefore,

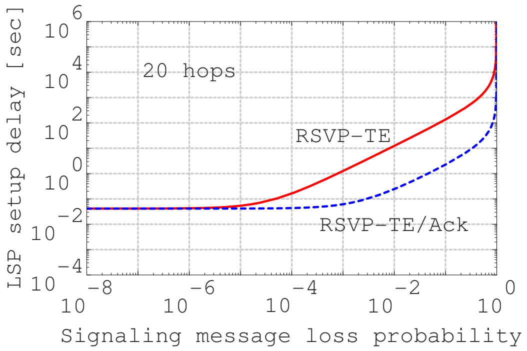
$$\lambda_1 \approx \frac{1}{T}, \tag{7}$$

$$\lambda_2 \approx \frac{1}{T} + \frac{1}{R}. \tag{8}$$

In [10], an RSVP–TE software module is implemented

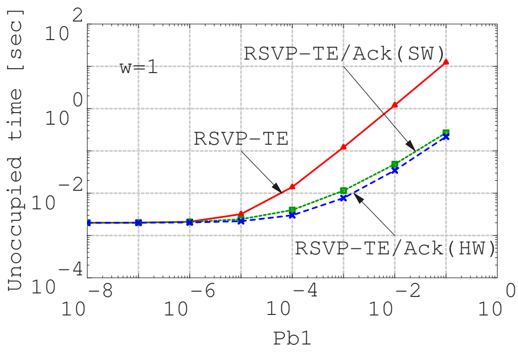


(a) 1 hop

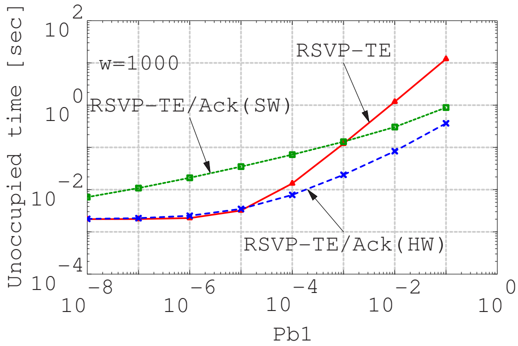


(b) 20 hops

Figure 7. Comparison of setup time between different lengths of LSP



(a) $w = 1$



(b) $w = 1000$

Figure 8. Effectiveness of message retransmission of RSVP–TE/Ack

and takes about 0.1 msec to process a signaling message. On the other hand, an RSVP–TE hardware module is implemented in [8] and it requires about 2.4 μsec to process a signaling message. We use these values for μ_p. Figure 8 illustrates the effectiveness of message retransmission, with the horizontal axes representing P_{b_1}, and the vertical axes representing the unoccupied time for a single–hop LSP. The unoccupied times of RSVP–TE/Ack are obtained with the model in Sec. 2 and B_2 that is calculated using Eqs. (5), (6), (7), and (8). The plots of RSVP–TE/Ack (SW) are the unoccupied times where the RSVP–TE module is implemented with software. RSVP–TE/Ack (HD) represents that the RSVP–TE module is implemented with hardware. RSVP–TE/Ack outperforms RSVP–TE regardless of the type of implementation when the number of sessions is one. However, when the number of sessions is 1000, the unoccupied time of RSVP–TE is shorter than that of RSVP–TE/Ack (SW) when the message loss probability in RSVP–TE is lower than 10^{-3}. This implies that the increase of the quantity of signaling messages due to message retransmission can result in poor resource utilization if the message loss probability is low.

6. Conclusion

In this paper, we developed a Markov model of GMPLS RSVP–TE for single–hop and multi–hop LSPs and analyzed the performance of variants of GMPLS RSVP–TE as well as backward reservation with the hard–state control. From the results, we demonstrated that resource utilization by RSVP–TE can be equivalent to that of a hard–state protocol when the loss probability of signaling messages is relatively low. The results regarding the performance analysis with control plane failure show that the hard–state signaling is unstable, and that is why the soft–state signaling is required for actual networks.

Message retransmission improves the responsiveness of GMPLS RSVP–TE when signaling messages are lost. However, it also increases the number of signaling messages and raises the probability of signaling message loss. We used the numerical results of our analysis to investigate the effectiveness of message retransmission, and found that the use of message retransmission can result in poor resource utilization. Specifically, when the signaling message loss probability is lower than 0.001 and when there are more than 1,000 LSP sessions, using message retransmission decreases the resource utilization of RSVP–TE if the RSVP–TE modules are implemented with software. Even if the RSVP–TE modules are implemented with hardware, this can be observed when there are more LSP sessions.

As for future research, we plan to analyze the performance of other signaling protocols for wavelength–routed networks, such as Parallel Reservation, and to compare the performance of soft–state and hard–state signaling protocols in the transient state.

Acknowledgments

This work was supported in part by a Grant-in-Aid for Young Scientists (A) 17680004 from the Ministry of Education, Culture, Sports, Science and Technology of Japan. The first author received support from the JSPS Research Fellowships for Young Scientists.

References

[1] L. Berger. Generalized Multi–Protocol Label Switching (GMPLS) signaling functional description. *RFC 3471*, Jan. 2003.

[2] L. Berger. Generalized Multi–Protocol Label Switching (GMPLS) signaling Resource ReSerVation Protocol–Traffic Engineering (RSVP-TE) extensions. *RFC 3473*, Jan. 2003.

[3] L. Berger, D. Gan, G. Swallow, P. Pan, F. Tommasi, and S. Molendini. RSVP refresh overhead reduction extensions. *RFC 2961*, Apr. 2001.

[4] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin. Resource ReSerVation Protocol (RSVP) – version 1 functional specification. *RFC 2205*, Sept. 1997.

[5] P. Ji, Z. Ge, J. Kurose, and D. Towsley. A comparison of hard–state and soft–state signaling protocols. In *Proceedings of ACM SIGCOMM '03*, Karlsruhe, Germany, Aug. 2003.

[6] K. Lu, J. P. Jue, G. Xiao, I. Chlamtac, and T. Ozugur. Intermediate–node initiated reservation (IIR): A new signaling scheme for wavelength–routed networks. *IEEE Journal on Selected Areas in Communications*, 21(8):1285–1294, Oct. 2003.

[7] R. Ramaswami and A. Segall. Distributed network control for optical networks. *IEEE/ACM Transactions on Networking*, 5(6):936–943, Dec. 1997.

[8] H. Wang, R. Karri, M. Veeraraghavan, and T. Li. A hardware–accelerated implementation of the RSVP–TE signaling protocol. In *Proceedings of IEEE International Conference of Communications (ICC 2004)*, volume 27, pages 1609–1614, Paris, France, June 2004.

[9] X. Yuan, R. Melhem, and R. Gupta. Distributed path reservation algorithms for multiplexed all–optical interconnection networks. *IEEE Transactions on Computers*, 48(12):1355–1363, Dec. 1999.

[10] Z. Zhou and D. Gao. An efficient adaptation of RSVP–TE in GMPLS. In *Proceedings of the 2004 Internatinal Symposium of Performance Evaluation of Computer and Telecommunication Systems (SPECTS 2004)*, pages 93–97, San Jose, California, July 2004.