Master's Thesis

Title

A Study on Routing Algorithms with Delayed Link State Information for Distributed Lightpath Network

Supervisor

Professor Masayuki Murata

Author

Takahiro Toku

February 15th, 2002

Graduate School of Information Science and Technology

Osaka University

Master's Thesis

A Study on Routing Algorithms with Delayed Link State Information for Distributed Lightpath Network

Takahiro Toku

Abstract

Wavelength division multiplexing (WDM) technology offers a large bandwidth for optical fibers to carry both today's Internet traffic and that of the future. Lightpath networks, where all–optical wavelength channels (*called lightpaths*) convey the traffic, have been considered to effectively utilize this WDM technology. There are two approaches to establishing lightpaths; the first is centralized and the second is distributed. In distributed lightpath networks, each node sets up and tears down the lightpath between a pair of communicating nodes. Therefore, conflicts about wavelength reservations during signaling operations may occur because of a lack of precise link–state information. A great deal of research has been devoted to avoiding conflicts to provide more efficient operating methods, i.e., reservation protocols, and routing and wavelength assignment algorithms. Previous studies on routing and wavelength assignment algorithms have assumed that global link–state information is provided without delays and the route is selected adaptively. However, in establishing lightpaths distributed way, if sender nodes adaptively select the route, the blocking probability may increase because of delayed link state information.

We therefore evaluated and found what effect delayed link–state information has on reservation protocols and routing algorithms through some simulations and implementation experiments. More specifically, we evaluated the average path setup time for routing algorithms in distributed networks, and found how the frequency of link–state information exchange affects the precision of collected link–state information. From simulation results, we conclusively confirmed that a backward reservation–protocol lessens the effect delayed link–state information has. The effect is less than 0.2% increase of blocking probability with the backward reservation protocol, while the forward reservation protocol increases the blocking probability more than 15%. We also confirm the proposed alternate routing algorithm shortens lightpath setup time about 20% than the least loaded routing algorithm assuming ideal conditions.

The simulation results includes some assumptions that did not reflect the actual processing overheads well. We therefore implemented a lightpath network conformed to the GMPLS standard and validate the actual computational overheads and simulation results. The results of the experiments proves the proposed routing algorithm performed as we had expected from the simulation results. We next investigate and evaluate the scalability of GMPLS based lightpath networks using the arrival rate at which the network saturates. From experiments, when the number of wavelengths is small, overhead of link–state–information exchange much affect the performance. Thus, we introduce a threshold–based link–state–information exchange method that improves the saturated arrival rate 10% higher than other conventional methods.

Keywords

WDM (Wavelength Division Multiplexing), lightpath network, backward reservation protocol, forward reservation protocol, adaptive routing algorithm, alternate routing algorithm, link state information, propagation delay, lightpath setup delay

Contents

List of Figures

List of Tables

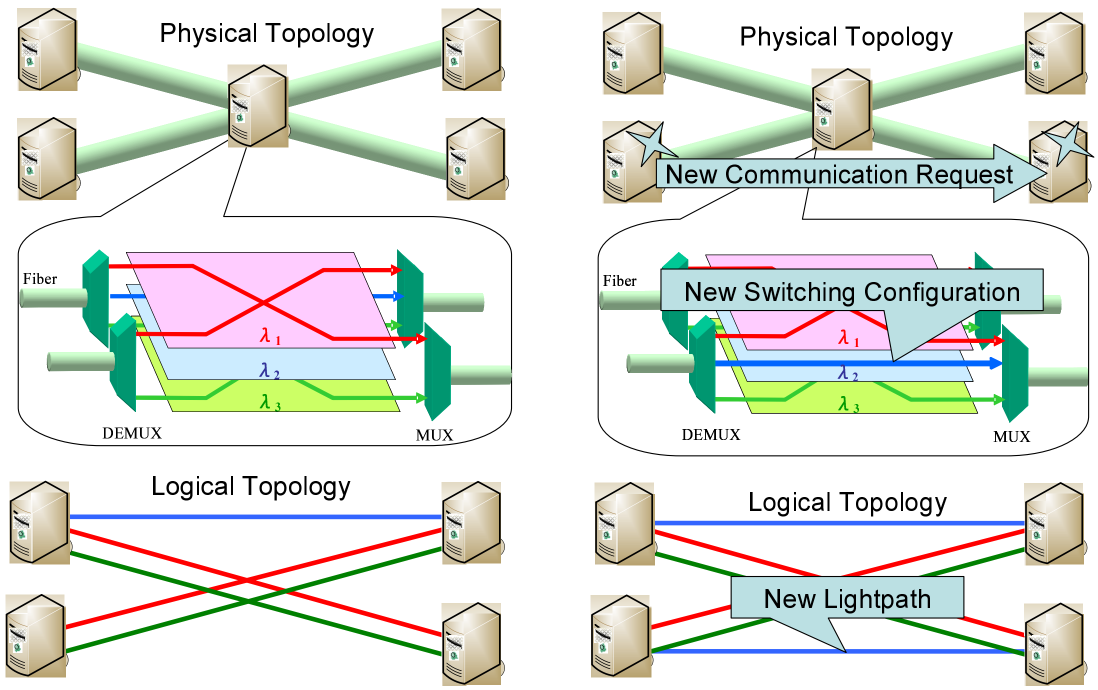
1 Introduction

Although bursty Internet traffic has been increasing because of P2P file–sharing and voice communications, Internet backbone networks lack the capacity for this growing traffic. Presently, wavelength division multiplexing (WDM) provides multiplex wavelength channels on a single fiber, enables high–capacity parallel transmission, and is expected to provide capacity for backbone networks.

One way of using WDM technology is to establish wavelength channels (called *lightpaths*) on a demand basis (Fig. 1(a)). A lightpath network consists of nodes with switching devices and links to optical fibers in a physical network. OXC is a switching device that binds an input wavelength channel to a specified output wavelength channel on the same wavelength. Lightpaths are formed through this switching process in intermediate nodes. When a new lightpath request arrives, the intermediate node switches specified fibers on a wavelength (Fig. 1(b)). When a data transfer request arrives at the sender node in a lightpath network on a demand basis, one wavelength is reserved along the route between the sender and the receiver nodes (Fig. 2(a)) [1, 2]. After data have been transferred along the lightpath, the wavelength is immediately released (Fig. 2(b)). As two or more lightpaths cannot share a wavelength on a fiber, some method is necessary to control the process of assigning routes and wavelengths in lightpath networks.

There are two approaches to establishing lightpaths. The first is the centralized approach, in which a special node sets up and tears down lightpaths. The special node manages all the lightpath requests, and therefore can select appropriate routes and wavelengths for lightpath requests. The disadvantage of this network is scalability, the network scale more grow, the lightpath set–up request more frequently arrives to the special node, which cannot process all of requests. The second is the distributed approach, in which each node can set up and tear down lightpaths. Because nodes do not know whether the other nodes are trying to reserve wavelengths in the distributed approach, conflicts about wavelength reservations may occur. To minimize the probability of such conflicts in establishing lightpaths, the sender node must select the appropriate route and wavelength for the lightpath. The nodes should know the state of wavelength utilization within the network to find and select which route is appropriate.

Each node needs precise information about the use of wavelength resources so that the routing algorithm can find the best route. In a distributed network, however, each node only knows about



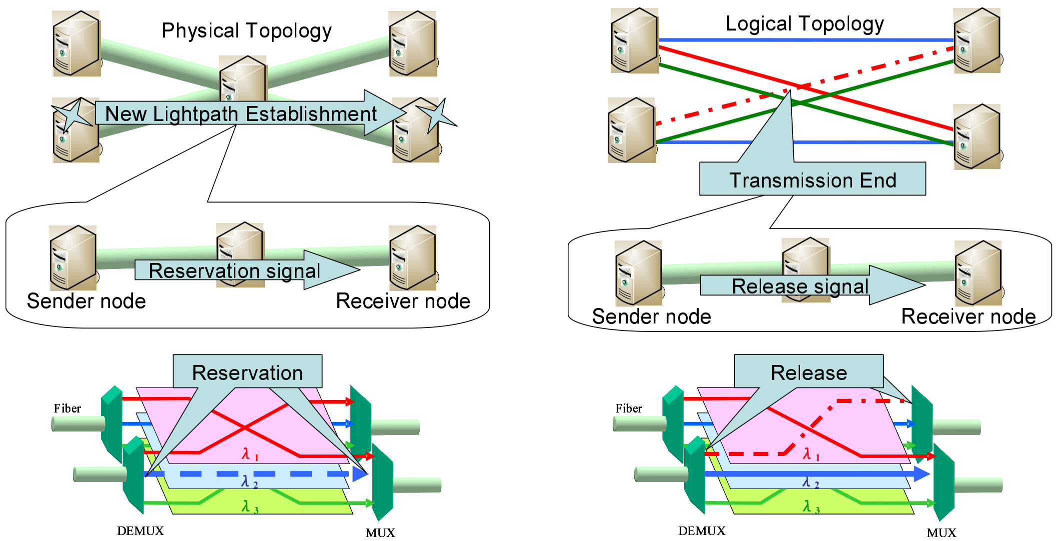
(a) Lightpath network

(b) New lightpath establishment

Figure 1: Lightpath network: physical topology, logical topology, and switching

the states of adjacent links, so they must exchange link–state information to efficiently select the appropriate route. There are two types of link–state information exchange. The first is frequent exchanges where nodes distribute link–state information immediately if the states of wavelength utilization change. The second is infrequent exchanges where nodes distribute link–state information periodically or when the states change over a given threshold. When nodes periodically exchange information, the amount of exchanged data is much smaller than with frequent exchange. However, the blocking probability may increase because of the discrepancy between the current status of wavelength use and the exchanged link state information [1]. Even if nodes exchange link state information every time the link state changes, propagation delays prevent this information from arriving at all the nodes at the same time, which affects route and wavelength selection at the sender node [3–5]. Furthermore, traffic for link–state information exchange much increases.

In [6, 7], a destination–node–oriented reservation protocol (backward reservation protocol) is described. The backward reservation protocol collects information about available wavelength resources during wavelength reservation, and the sender node only needs to select the route. There-



(a) Wavelength reservation to establish a lightpath

(b) Wavelength release for tear down a lightpath

Figure 2: Lightpath establishment and release

fore, because the necessary information for all nodes is about route selection, there is no need for frequent link–state information exchange and detailed link state information. For example, link–state information only about the use of wavelength resources is sufficient to select a route.

Many routing algorithms have been studied for lightpath networks [1, 2, 8–11]. Two main algorithms have previously been proposed for routing lightpaths: adaptive–routing and alternate–routing algorithms. With the first, a sender node, at which a lightpath–setup request has arrived, evaluates all available routes in the network, according to the current status of wavelength utilization, and selects the one that will provide the best route for the lightpath. With the second, each node has a route–list in which a set of pre–defined routes is described. The routes in the list are ordered by hop–counts, for example, and the sender node selects a route from the list. If the lightpath setup on the selected route fails, the sender node tries the next route. Although adaptive routing has better performance than alternate routing [12], it requires additional overheads to calculate appropriate routes from link–state information. Alternate routing requires less computational complexity than adaptive routing since the set of routes is pre–defined and no routes need to be calculated when the lightpath–setup request arrives at sender nodes. However, the discrepancies between the current status of wavelength use and the exchanged link–state information have not been considered in these studies. These discrepancies make sender nodes select the "worse" route, which increases the probability of requests being blocked. Routing with the discrepancy is worse than that without it. Therefore, the adaptive routing algorithms do not perform well since periodic information exchanges and the above–mentioned propagation delays prevent link–state information from being precise. If sender nodes select a route adaptively, the blocking probability is considered to increase because of the delayed link–state information. With the alternate routing algorithm, on the other hand, the sender node selects the next route from pre–defined routes if the path setup fails. We found the degradation in performance is small with alternate routing. In this paper, we discuss our evaluation of the average path setup time for routing algorithms to clarify the effect delayed link state information has. Currently, GMPLS (Generalized Multi-protocols Label Switching) has achieved a common control plane for optical and electronic data transmission in the Internet Engineering Task Force (IETF). The GMPLS architecture [13] provides a unified control plane for networks, including both packet switching and circuit switching technologies. GMPLS defines a layered hierarchy for switching capabilities, i.e., packet switching, layer–2 switching, TDM (Time Division Multiplexing), lambda switching, and fiber switching capabilities. To bundle and manage the links of different switching types, GMPLS has a generalized label and interface switching capabilities. GMPLS proposes extensions to the open shortest path first for traffic engineering (OSPF–TE) [14] and the reservation protocol extended for traffic engineering (RSVP–TE) [15]. The reservation signaling protocol, RSVP–TE, configures and controls generalized label switched paths. This includes the distribution of link–local labels, which identify resources on links between neighboring nodes, and configure switching fabrics in nodes along the path. The OSPF–TE protocol provides for the distribution of link–state information, including traffic engineering (TE) attributes (e.g., available bandwidth and available wavelength) of switching capable interfaces. RSVP–TE uses the route obtained from this TE information for the reservation. GMPLS also defines the link management protocol (LMP), which manages the control plane to detect link failures and node problems. This protocol provides the managed control channel to other control plane signalings [16, 17].

Therefore, a lightpath network is achieved in GMPLS on switching wavelength labels (3). There is an example of lightpath networks with GMPLS in Fig. 3. Each label switching router can switch labels assigned a wavelength. The core node has a label switching table that determines

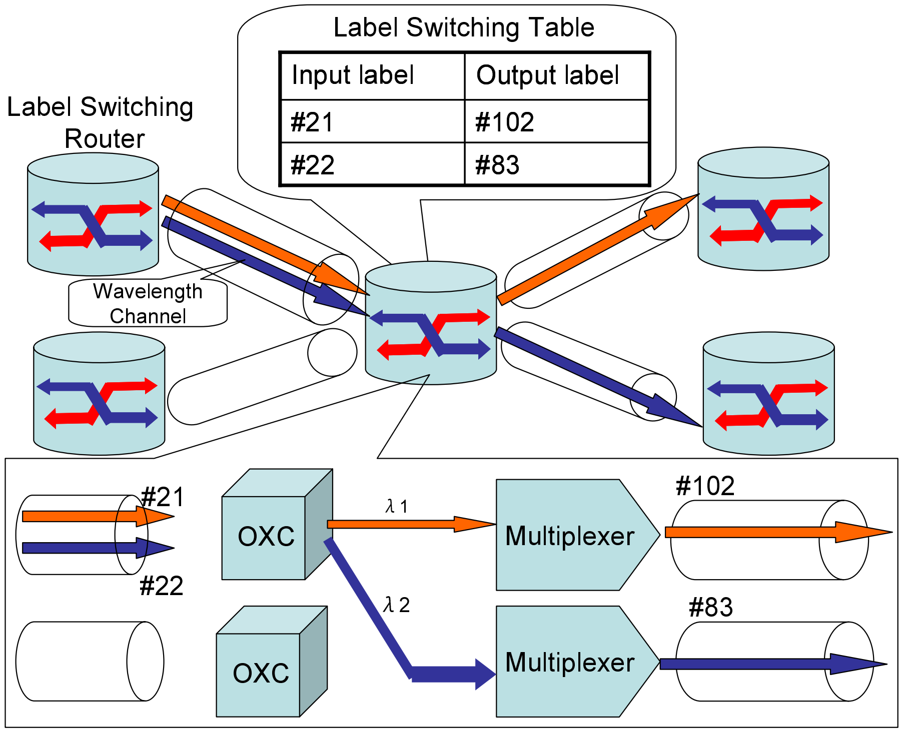


Figure 3: GMPLS label switched lightpath network

the binding of wavelength channels. Label #21 is assigned a λ_1 wavelength channel on an input fiber, and label #102 is assigned a λ_1 wavelength channel on an output fiber. That is, one lightpath consisting of two wavelength channels is established. GMPLS only specifies the signaling standard in the control plane except for the route selection algorithm, resource reservation scheme, and link–state–information exchange. We will discuss these components of a lightpath network from the viewpoint of delayed link–state information. Furthermore, we implemented components, resource reservations, and link–state–information exchange, conforming to the GMPLS standard, which we will also discuss.

This thesis is organized as follows. In Section 2, we explain the existing routing and wavelength selection methods and wavelength reservation protocols. In Section 3, we discuss how the frequency of link–state–information exchange affects the establishment of lightpaths using computer simulations. In Section 4, we explain how we implemented and validated the results obtained from simulations using those from experiments. The conclusion is in Section 5.

Table 1: Reservation protocols: operations at nodes

Node	Forward reservation protocol	Backward reservation protocol
Sender node	route and wavelength selection and reservation	route selection and PROBE signaling
Upstream intermediates	reservation	update PROBE signal
Receiver node	return ACK	wavelength selection and reservation
Downstream intermediates	forwarding ACK signal	reservation

2 Routing and Wavelength Assignment in Optical Networks

There are two elements involved in establishing lightpaths: route and wavelength selection and the reservation protocol. One of the most important issues facing routing in distributed networks are the intervals between link–state–information exchanges. If nodes exchange link–state–information every time the link status changes, huge amounts of information spread throughout the network and the routing table is frequently calculated making increasing the loads of CPUs in nodes. To reduce these processing overheads, some method is needed that enables less frequent link–state exchanges using less detailed link–state information.

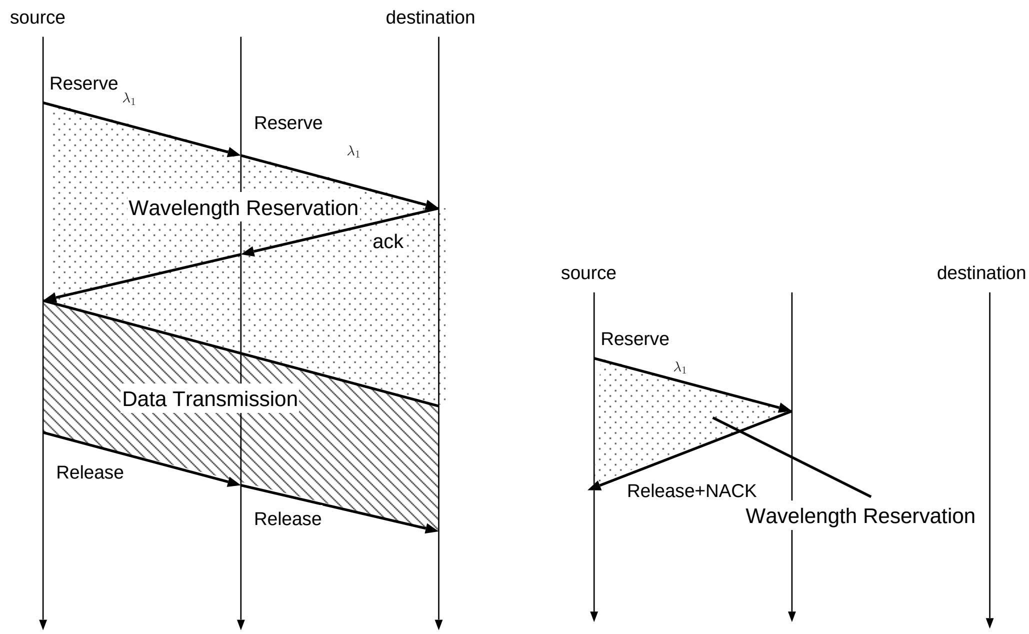
We will first explain the wavelength reservation protocol in Subsections 2.1. We will then discuss the routing algorithms, i.e., fixed–routing, adaptive–routing and alternate–routing algorithms, in Subsection 2.2.1, 2.2.2 and 2.2.3.

2.1 Wavelength reservation protocols

There are two reservation protocols; forward–reservation (2.1.1) and backward–reservation (2.1.2). Table 1 summarizes the operations of nodes on the lightpath set–up.

2.1.1 Forward reservation

When a request to establish a lightpath arrives at the sender node, it selects the route and wavelength for the lightpath. The sender node then transmits a RESERVE signal and reserves the wavelength along the selected route. When an intermediate node receives the signal, it obtains the wavelength from the signal, and reserves the wavelength on the next link. When the RESERVE



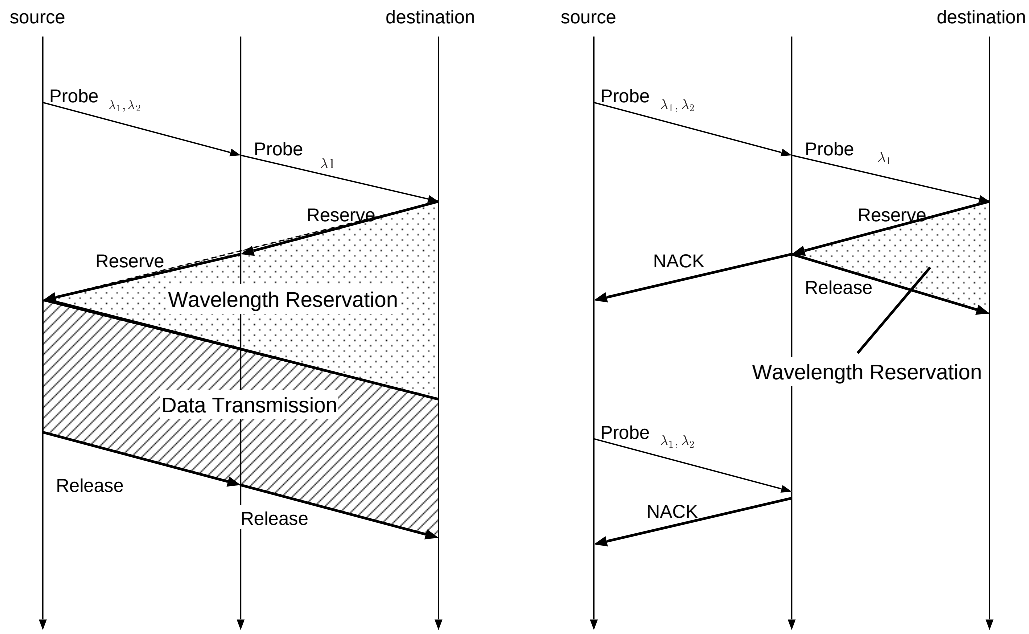
(a) A successful case of a lightpath establishment

(b) A failure case of a lightpath establishment

Figure 4: Forward reservation protocol

signal arrives at the receiver node, a lightpath is established and the receiver node transmits an ACK signal to the sender node (Fig. 4(a)). The sender node transfers the data upon receiving the ACK signal, and transmits a RELEASE signal to the receiver node at the end of the data. The RELEASE signal releases the wavelength used for the lightpath. Figure 4(b) illustrates a case where lightpath establishment has failed. The RESERVE signal has arrived at the intermediate node, but the wavelength is already reserved or is being used by another lightpath. Here, the request to establish a lightpath has been rejected, and the intermediate node transmits a NACK signal to the sender node.

The forward reservation protocol needs to select a route as well as a wavelength at the sender node; the link state information should include information about the use of each wavelength on each link. We can use the number of available wavelengths as link state information. However, the sender node may select the wrong wavelength because of this less–detailed link state information, and blocking probability will increase.



(a) A successful case of lightpath establishment

(b) A failure case of lightpath establishment

Figure 5: Backward reservation protocol

2.1.2 Backward reservation

When a lightpath request arrives at the sender node, it only selects the route for the lightpath. The sender node then generates a PROBE signal containing a set of available wavelengths on the next link, and transmits this to the receiver node. When an intermediate node receives the PROBE signal, it intersects the sets of available wavelengths on the next link that are contained in the PROBE signal, and writes in the PROBE signal.

After updating the PROBE signal, the node transmits the signal to the next node. The set of wavelengths in the PROBE signal contains available wavelengths on the route along which the PROBE signal arrived at the receiver node. The receiver node selects a wavelength from the available wavelengths in the PROBE signal, and transmits a RESERVE signal to reserve the wavelength on the route. On receiving the RESERVE signal at the sender node, the sender node acknowledges that a lightpath has successfully been established, and starts transferring the data. After the data have been transferred, the reserved wavelength is released via a RELEASE signal.

Table 2: Comparison of routing algorithms: advantages and disadvantages

Routing Algorithm	Advantages	Disadvantages
Fixed routing	No CPU consumption of route selection	Blocking increase because of convergence
Adaptive Routing	avoid highly–loaded route	need link state information exchange
Alternate Routing	can retry other routes if establishment fails	may result in wrong route selection

Figure 5(a) illustrates a case of successful wavelength reservation. There are two instances when a request to reserve wavelength can be rejected with the backward reservation protocol (Fig. 5(b)). The first is when the available wavelengths are being probed (PROBE sequence), and the second is when the wavelength has already been reserved (RESERVE sequence).

Rejection on receiving a PROBE sequence occurs when the set intersected by the intermediate node is empty. When this happens, there are no available wavelengths on the route, and the intermediate node sends a NACK signal to the sender node. Rejection on the receipt of a RESERVE sequence occurs when wavelength reservation conflicts with the establishment of another lightpath. When wavelength reservation fails, a NACK signal is transmitted to the sender node, and a RELEASE signal is transmitted from the intermediate node to the receiver node to release the reserved wavelength.

2.2 Routing algorithms

Routing algorithms select routes for path setup. They use link–state information and candidate routes. In this subsection, we explain three typical routing algorithms, and their advantages and disadvantages. Table 2 briefly outlines routing algorithms.

2.2.1 Fixed routing

The fixed routing algorithm uses a predetermined route for all node pairs each time a connection is established [11, 18]. On the arrival of a lightpath request, the fixed routing algorithm chooses a predetermined route. Route selection does not depend on the actual dynamic link–state change.

A typical fixed routing algorithm is minimum hop routing (for instance, Dijkstra's shortest path algorithm). When lightpath establishments are congested on certain links, this algorithm cannot reroute the link, and the lightpath request may be blocked.

In this paper, we discuss our use of a minimum hop routing algorithm, where the sender node selects the shortest route to connect a node pair.

2.2.2 Adaptive routing

With adaptive routing, sender nodes dynamically select a route to the receiver node when a lightpath setup request arrives ([1, 10, 11, 19]). Route selection depends on network–state information, i.e., both the connectivity of each adjacent node and the wavelength utilization of each link. The advantage with this algorithm is that the sender node evaluates all available routes in the network, which is expected to result in less blocking. The disadvantage with this algorithm is that the discrepancy between the current status of wavelength utilization and exchanged link–state information greatly affects blocking. With this algorithm, it is necessary to efficiently select the proper route that current link–state information can use.

In this research, we used a least loaded routing algorithm, where the sender node selects the route that has the minimum number of reserved wavelengths along it. Note that the least loaded routing algorithm requires the number of reserved wavelengths in each link as the link state information.

2.2.3 Alternate routing

There are two types of alternate routing; fixed–alternate and adaptive–alternate routing [12,20–22]. With fixed–alternate routing, each node has a route–list for a set of pre–determined routes. This list contains an ordered list of routes to each destination node, and the routes are not changed dynamically. When a lightpath setup request arrives at the sender node, the node selects a route (primary route) according to its order on the list. If the lightpath cannot be established along the primary route, the sender node then selects the next route. This continues until all the routes in the list have been examined. An advantage of fixed–alternate routing is that since the list has been determined in advance, the route does not have to be calculated before the lightpath has been set up. Furthermore, even if some links fails, the sender node can easily select other routes. With adaptive routing, the sender node must calculate another route to avoid failed links.

With adaptive–alternate routing, each node also has a route list, but the order of routes changes dynamically according to wavelength utilization in the network. This is a hybrid approach to adaptive routing and fixed–alternate routing to balance the number of wavelengths used in each link while providing less computational complexity in selecting routes. If the sender node is aware of congestion on links based on link–state–information exchange, it establishes an order to load–balance the network. If the lightpath setup along the route fails, the sender node selects another route by considering whether the link load on the route would be lower. As the order of routes can be changed by the sender node with adaptive routing, the degradation in performance due to delayed information was considered low using adaptive routing with an appropriate routing algorithm.

In following sections, we discuss our evaluation of the routing algorithms and reservation protocols from the viewpoint of degradation in performance due to delayed link–state information.

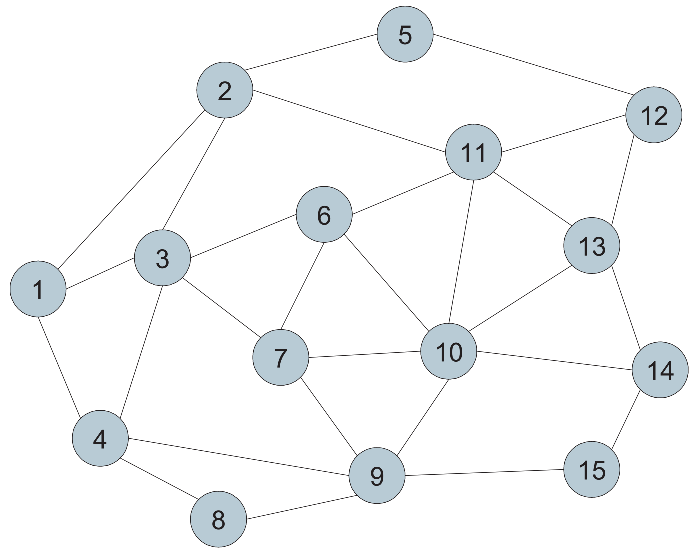


Figure 6: Random network model

3 Performance Evaluation

This section discusses our evaluation of reservation protocols and routing algorithms in distributed environments using computer simulations.

3.1 Simulation model

Figures 6–7 have the two network topologies we used to evaluate performance. Figure. 6 has a random network consisting of 15 nodes and 28 duplex links. There is an average number of 2.50 minimum hop–counts between node pairs, and the mean propagation delay for each link was set by multiplying the length of each link in Fig. 6 by scale factor α. Figure 7 shows the Japan backbone network, which consists of 49 nodes and 91 duplex links. The average number of minimum hop–counts is 6.06 in this network, and the mean propagation delay is 0.59 ms.

We did simulations on computer with the following parameters.

- Requests arriving at each node follow with the Poisson arrival with a mean of P. The arrival rate to each node pair is even.
- The service time for a lightpath has an exponential distribution with a mean of $1/\mu$.
- The number of multiplexed channels in each optical fiber is $W + 1$. One channel is used as a control channel on which the nodes exchange control signals and link–state information.

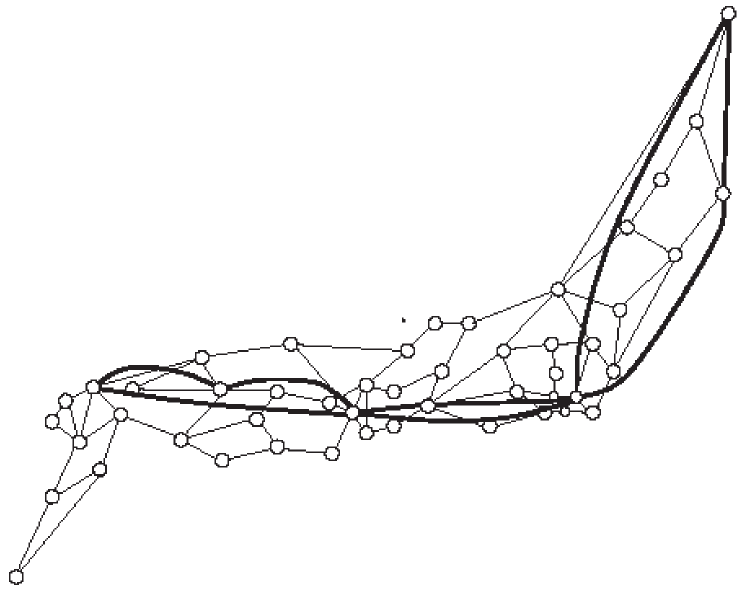


Figure 7: Japan backbone network

The other W channels are used to establish lightpaths.

- Link–state information is updated at T intervals.

We assumed that there would be no processing delays in the routing, wavelength–selection, or wavelength-reservation processes at each node. Every control signal is delayed due to the effect of link propagation delay. The signals are not affected by either node processing delays or queueing delays.

3.2 Routing algorithms

Let us next discuss our evaluation of adaptive routing and alternate routing performed. We used the backward reservation protocol to select an available wavelength on the route. We also used the k–shortest path algorithm to select a route for the lightpath ([23]). With adaptive routing, the best route (i.e., the least loaded route of the k–shortest routes) is selected by the sender node. In alternate routing, the sender nodes decide the order the routes will be selected from the k–shortest paths. If a lightpath is set up k–times, the sender node re-determines the order routes will be selected from the k–shortest paths.

Table 3: Alternate routing algorithms: routing type and brief summary of selected route

Algorithm name	Routing Type	Selected route (brief summary)
Shortest Path	Fixed	minimum hop count
FAR	Alternate	By order of hop count
Least Loaded	Adaptive Alternate	Least loaded route is selected
FAR with LL	Adaptive Alternate	By order of load
FAR with 1SP and LL	Adaptive Alternate	Primary route is shortest, others are in order of load.

Table 3 summarizes the routing algorithms we used in our simulations. With the shortest path (SP) algorithm, the sender node repeatedly selects the shortest path that has the minimum hop counts. With FAR (: Fixed Alternate Routing), the sender node only selects routes according to their defined order. Each node selects a route from the route–list in ascending sequence with regards to the number of hop–counts. With the least loaded algorithm, the sender node selects the least loaded route according to defined routes and collected link–state information that has been dynamically updated. If path setup fails, the sender node selects the least loaded route using link–state information that has been collected at that time. As this algorithm balances the number of reserved wavelengths on the links, the blocking probability is low in highly–loaded networks.

The “FAR with LL” is a fixed alternate routing algorithm with adaptive (least loaded) selection. The sender node selects a primary route as with the least loaded algorithm and the other defined routes are sorted by order of load. If path setup fails, the sender node selects the next route from the left of the sorted routes. This algorithm also balances the number of reserved wavelengths on the links. Routes are always selected using delayed link–state information with the least–loaded algorithm, which degrades performance. However, with the “FAR with LL” algorithm, as the sender node can select other routes from candidate routes, the degraded performance of delayed link–state information is expected to be low.

Finally, let us introduce our new algorithm: “FAR with 1SP and LL” algorithm. With this algorithm, the sender node selects the less loaded route from candidate routes the same as the “FAR with LL” algorithm, except that the primary route is fixed to the shortest path. “FAR with 1SP and LL” algorithm has the advantages of “Shortest Path and FAR with LL” algorithm. The primary route selection with this algorithm consumes fewer wavelength resources, and if the setup fails, the next route is selected to achieves load–balancing.

3.3 Policies for link–state–information exchange

Routing algorithms use link–state information to select the proper route depending on the network actual state of the network. The initial exchange of link state information allows routers to build an incremental view of the network topology. The routers then monitor adjacent links, and exchange link attributes to neighbor node on some triggers of flooding. These triggers have been considered in [24].

(1) frequent exchange: flooding every time the state changes,

(2) periodic exchange: flooding every a few seconds.

Frequent exchange improves preciseness of exchanged link–state information. Therefore, routes can be properly selected depending on how accurate this information is. However, there is a huge amount of traffic from this link–state–information exchange because frequent link–state updates cause control channel congestion and heavy CPU calculations on flood the information. Therefore, we need to consider the trade–off with this triggering method between the control–channel and degradation routing granularity. Periodic exchange, on the other hand, degrades the accuracy of exchanged link–state information. How well route selection works depends on how much information has deteriorated. If the route has been selected adaptively, it needs to avoid congested links, but in this case the delayed link–state information prevents the route from being properly selected. However, the traffic from OSPF of periodic exchange is much smaller than that of frequent exchange. Therefore, we need to consider the trade–off between the frequency of link–state–information exchange and route selection accuracy. Ref [3] clarified the issues with this problem and proposed new link–state triggering methods. The nodes have a certain constant threshold for triggering and flood whenever the triggering value exceeds the given threshold. One of triggering value is the difference of current link state and previously flooded link state information and the other is relative proportion of current and previous link state. Flooding traffic and the effect of delayed link–state information depends on the threshold, and they found through evaluation that new methods provide better routing accuracy with less flooding than periodical link state exchange. However the OSPF standard does not support the new method. Therefore, we evaluated the periodic link–state exchange and frequent link–state exchange using computer simulation.

3.4 Evaluation of wavelength reservation protocols

The performance measurement for the reservation protocols is blocking probability. The blocking probability is the probability of lightpath establishment will fail. We used a model for this evaluation where the nodes terminated a lightpath establishment request if the lightpath could not be established. On the other hand, the performance measurement for the routing algorithms is lightpath setup delay. Lightpath setup delay is defined as the time from when a lightpath setup request was received at the sender node to when the lightpath was successfully established. We used a model where the nodes continued to establish a lightpath even if lightpath establishment had failed.

Route and wavelength selection algorithms for the forward and backward reservation protocols will now be described. With both forward and backward reservation protocols, the least loaded route is selected from the k–shortest paths. The least loaded route is defined as a route such that the maximum number of wavelengths used in each link on the route is minimal among k-shortest paths.

With the forward reservation protocol, the sender node selects a route with at least one available wavelength. If there are two or more available wavelengths, one is randomly selected. Note that information about wavelength use in each link is distributed as link state information. With the backward reservation protocol, the sender node selects the least loaded route from k–shortest paths. The receiver node then randomly selects a wavelength from the set of available wavelengths in the PROBE signal as described in Sec. 2.1.2. Note that information about the number of wavelengths used in each link is distributed as link state information with this protocol.

From Figs. 8 through 11, we can see the blocking probability for a lightpath request for different link state update intervals with both forward and backward reservation protocols. The x–axis is the arrival rate for a lightpath request and the y–axis is the blocking probability for a lightpath request. Both the x–axis and y–axis are log–scale. "Global" means that the sender nodes obtain global link state information, assuming that all nodes exchange link–state information with no propagation delay, which is an ideal case. Here, "T=0" means that the link state information is exchanged immediately after there has been a change in the link state, "T=15sec" means that link state information is exchanged every 15 seconds.

Figure 8 plots the blocking probability at arrival rate P. The number of multiplexed channels (W) was set to 8 and the average service time ($1/\mu$) was set to 1.0ms. The average link propagation delay was 0.1ms (α = 0.0557 ms). The results of blocking probability for "global" are almost the same as for "T=0" with both forward and backward reservation protocols. This is because the average link propagation delay is short, and the link–state information is transmitted with smaller delays. If we compare the results for "T=15sec" and those for "T=0", the blocking probability increases with both forward and backward reservation protocols. The results for the backward reservation protocol have a smaller increase than for the forward reservation protocol. The reason is that when link–state information is exchanged periodically with the forward reservation protocol, the probability that the route and wavelength selected by the sender node have already been reserved increases because the wavelength is selected based on the old link–state information. With the backward reservation protocol, the difference in blocking probability between link–state information and with the actual link state is small because the PROBE signal dynamically collects information about the available wavelengths on the route. Therefore, the blocking probability slightly decreases with the backward reservation protocol. Note that when the arrival rate is low (lower than 0.004), there are no significant differences between the results for "T=0" and "T=15sec" due to less frequent link–state–information exchanges.

Figure 9 plots the blocking probability when the average service time is 100ms. We can see that a longer service time significantly increases the blocking probability based on the difference between "T=0" and "T=15sec" with both forward and backward reservation protocols. When this occurs, lightpaths are held longer than in other situations, but the link–state information intervals are longer than the mean service time. Because the received link–state information often fails to reflect the actual link state, the selected wavelength is likely to have already been reserved for other lightpaths with the forward reservation protocol. With the backward reservation protocol, a long service time affects the PROBE sequence because the available wavelengths do not change often. Discrepancies between wavelengths available in the PROBE signal and actually available wavelengths do not occur frequently and RESERVE sequence are rejected less frequently. Therefore, rejection with PROBE sequence are dominant in blocking under these conditions. Rejection when a PROBE signal is received occurs when there is a discrepancy between the selected route and actual available wavelengths in routing at the sender node. Therefore periodic link–state exchanges affects blocking probability. Thus, both forward and backward reservation protocols need precise link–state information when the service time is long. However, blocking probability with

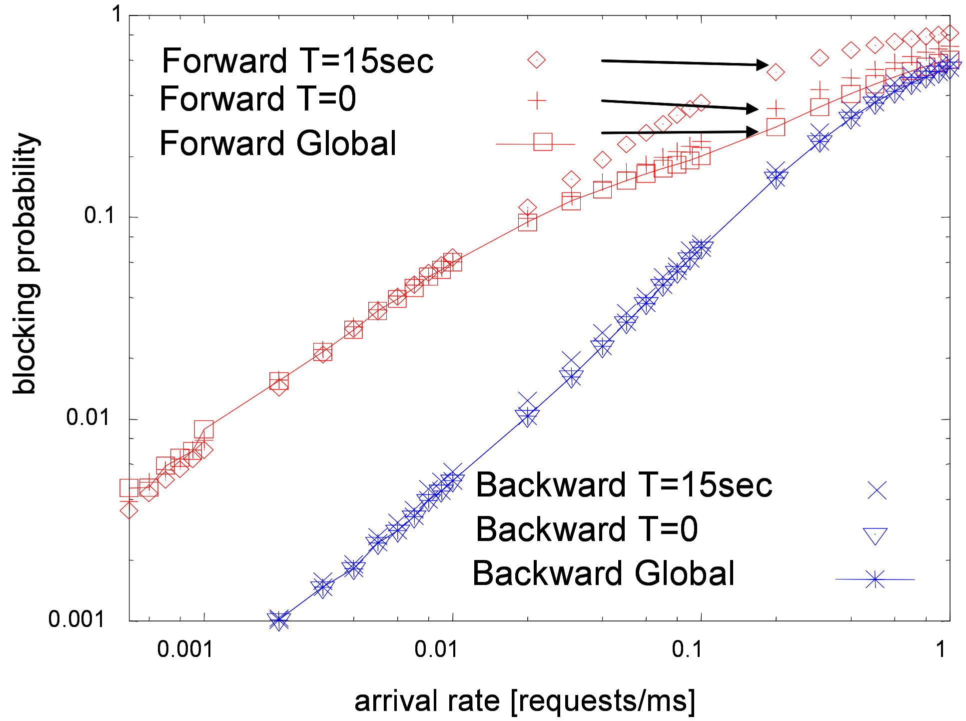


Figure 8: Blocking probability and link state update interval in random network : $W = 8$, $1/\mu =$ 1.0ms, average link propagation delay of 0.1ms (α = 0.0557ms)

the backward reservation protocol is small because the available wavelength collected based on the PROBE sequence works well.

We next examine both forward and backward reservation protocols on the Japan backbone network. Figure 10 plots blocking probability depending on the arrival rate. The advantage of frequent link–state exchange decreases both for forward and backward reservation protocols, compared with Figure 8. That is, the difference between the results for "T=0/15sec" and "Global" is not as great. This is because the minimum number of hop–counts between nodes for the Japan backbone network is larger than that for the random network. Lightpath establishment with larger hop–counts frequently causes conflicts. Here, the available wavelengths the sender node receives from link–state information tend to be reserved by other lightpath requests. Therefore, the advantage of routing with accurate link–state information is reduced, and blocking probability cannot be improved with frequent link–state exchanges.

We now explain how the link–state information interval affects wavelength selection by using a three–node tandem network for the simulation topology. The route for the lightpath is fixed, so we

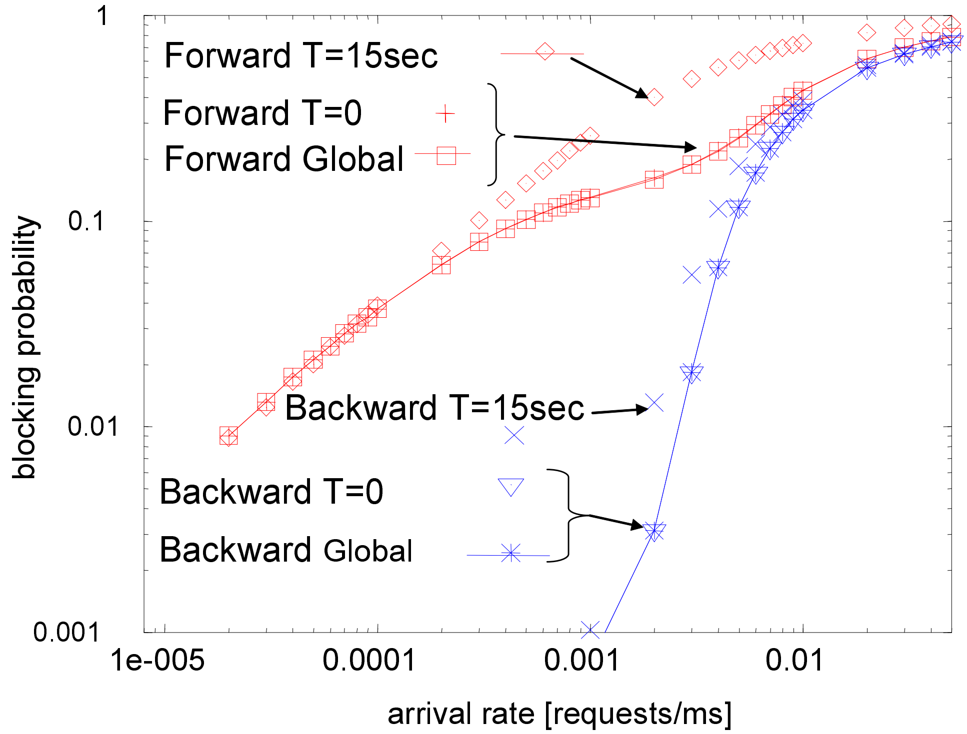


Figure 9: Blocking probability and link state update interval in random network : $W = 8$, $1/\mu$ = 100ms, average link propagation delay of 0.1 ms (α = 0.0557ms)

can only see the effect of discrepancy between the actual utilization of wavelength resources and the link–state information exchanged in wavelength selection. Figure 11 plots blocking probability dependent on the arrival rate. Interval T was set to 0 for this simulation because the backward reservation protocol does not use link–state information at the sender node to select wavelengths We can see that as the link–state–information–update interval increases, so too does the blocking probability. A shorter link–state update interval is needed with forward reservation protocol to select wavelengths. In contrast, with the backward reservation protocol, wavelength selection depends on the link state information, and a shorter interval is not necessary.

3.5 Evaluation performance of routing algorithms

From Figs. 12 through 15, we can see the average setup time for lightpath requests for different alternate routing algorithms with the backward reservation protocol. The x–axis is the arrival rate, and the y–axis is the average path setup time for a lightpath request. Both x– and y–axes are

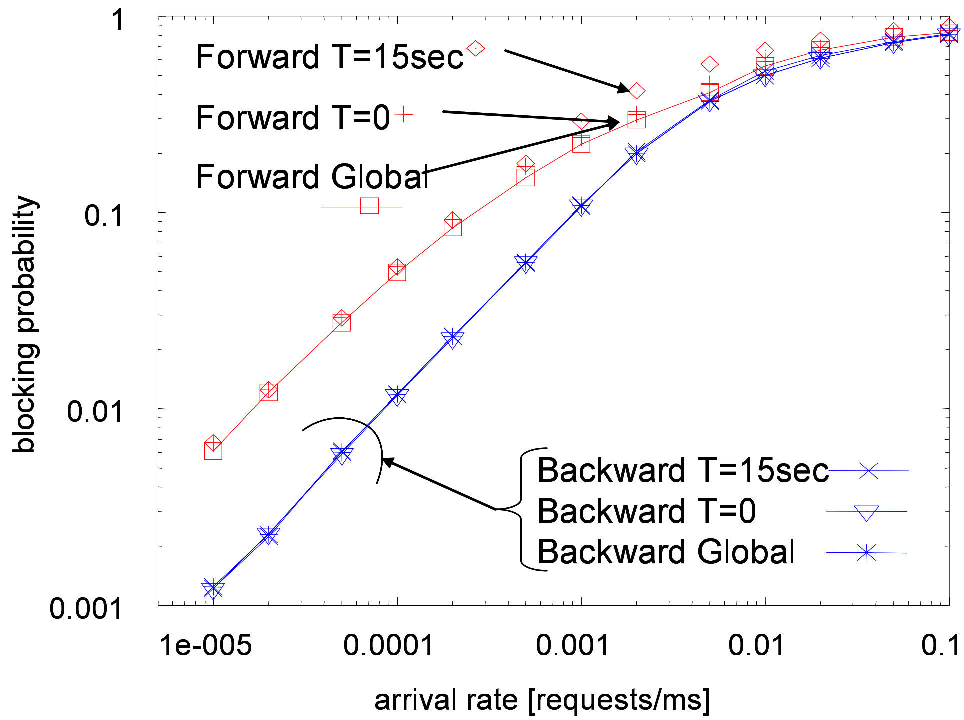


Figure 10: Blocking probability and link state update interval in Japan backbone network: $W = 8$, $1/\mu$ = 1ms, average link propagation delay of 0.59 ms

linear–scale. "Global" means that the sender nodes can use global link–state information without any propagation delay, which is an ideal case. Here, "T=0" means that link–state information is exchanged immediately after there has been a change in the link state ;"T=15sec" means that link–state information is exchanged every 15 seconds.

Figure 12 plots the average path setup time for algorithms with a random network topology (Fig. 6). We can see that the average path setup time with the "Shortest Path", with "FAR with 1 SP", and with "Least–Loaded Global" is shorter than that with the "Least–Loaded and FAR with LL" algorithm when the arrival rate is low. We can also see the slope for the average path setup time with the "Shortest Path" increases more steeply than with the other algorithms when the arrival rate is high. This is because when the arrival rate is low and the number of hop–counts of established lightpaths is small, more available wavelengths are left in the network than when the least–loaded route is selected. However, when the arrival rate is high, some links are over–loaded with the Shortest Path algorithm. Consequently the network is not load–balanced by the Shortest Path, so the average path setup time increases sharply as the arrival rate increases. However, the

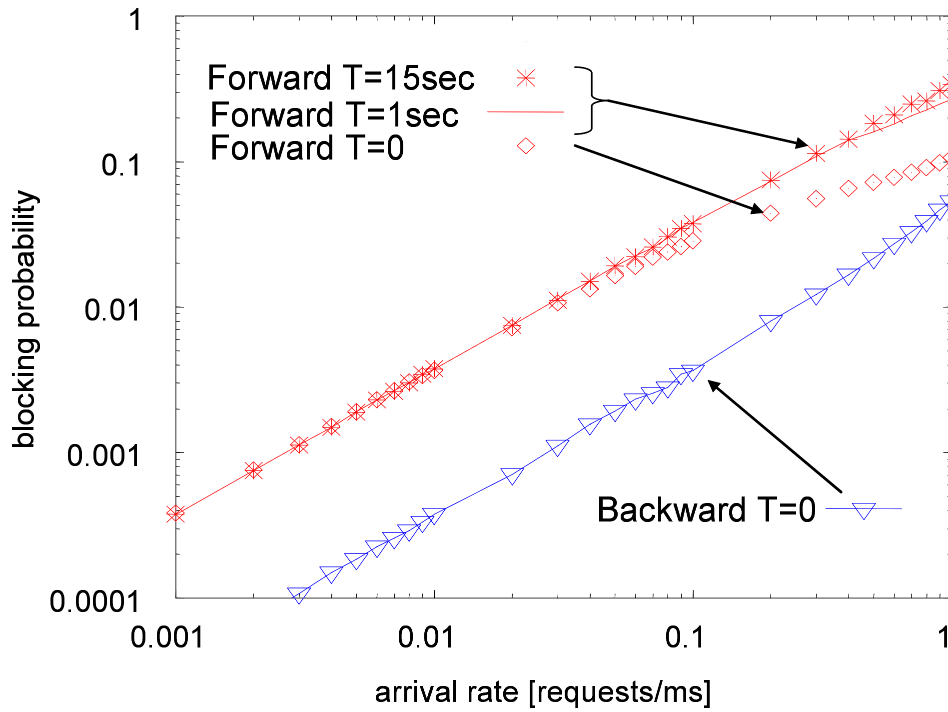


Figure 11: Blocking probability in three node tandem network $W = 8$, $1/\mu$ = 1.0ms, average link propagation delay of 0.1ms

load for all links are balanced with the other algorithms, so that the slope for the average path setup time does not exhibit the sharp increase.

Figure 13 compares the performance of algorithms with immediate link–state–information exchange and periodic link–state–information exchange. We can see that the difference between the results with "T=0" and "T=15sec" is small. That is, the discrepancy does not affect blocking. The main reason is that the backward reservation protocol determines the lightpath's wavelength based on information collected from available wavelengths on the route.

Figures 14–15 plot the performance of routing algorithms with the Japan backbone network topology. The differences in performance for each algorithm have a similar tendency to those in the random network. However, Fig. 15 does not plot the same as in Fig. 13; the performance of the least loaded algorithm is much more degraded than that of the "FAR with 1SP and LL" algorithm. This is because the average distance between nodes in the Japan backbone network is longer than that in the random network. That is, the inaccuracy of link–state information increases. In all the figures, "FAR with 1SP and LL algorithm" performs better than the other algorithms.

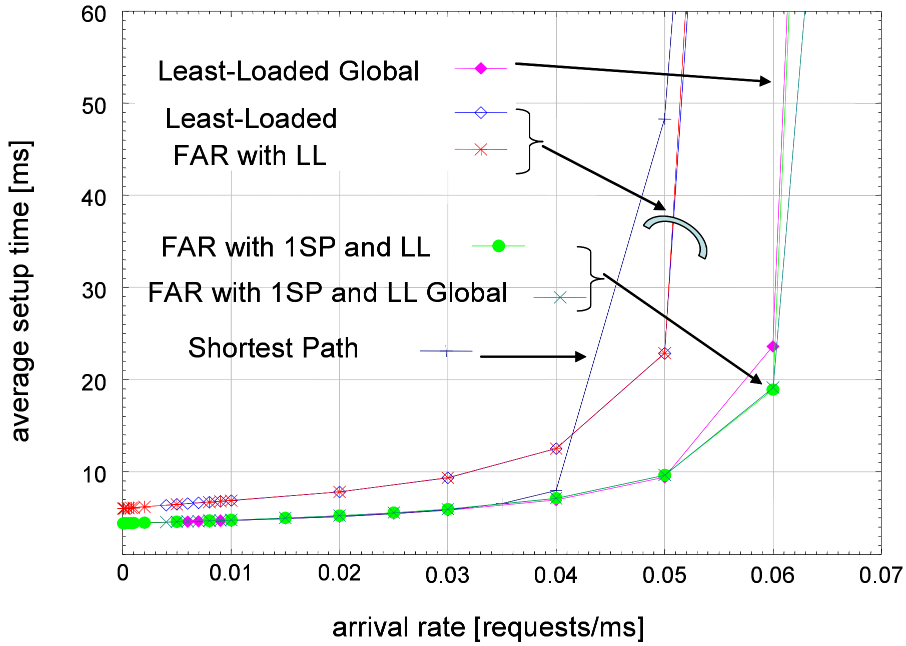


Figure 12: Average path setup time for random network with dynamic link use information: $W = 8$, $1/\mu$ = 1.0ms, average link propagation delay of 0.1 ms

It is important that routing algorithms select the route that uses the lowest amount of wavelength resources as well as one that balances the number of wavelengths used in each link. The "FAR with 1SP and LL" algorithm does both. Furthermore, its selecting of alternate routes reduces performance degradation due to imprecise link state information.

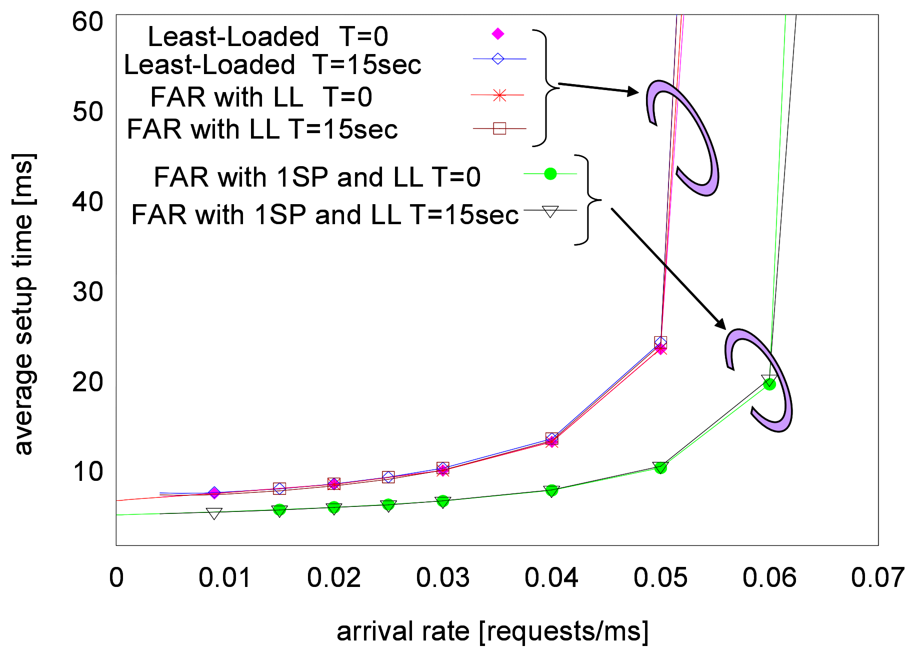


Figure 13: Average path setup time for random network with periodic link use information: $W = 8$, $1/\mu$ = 1.0ms, average link propagation delay of 0.1ms

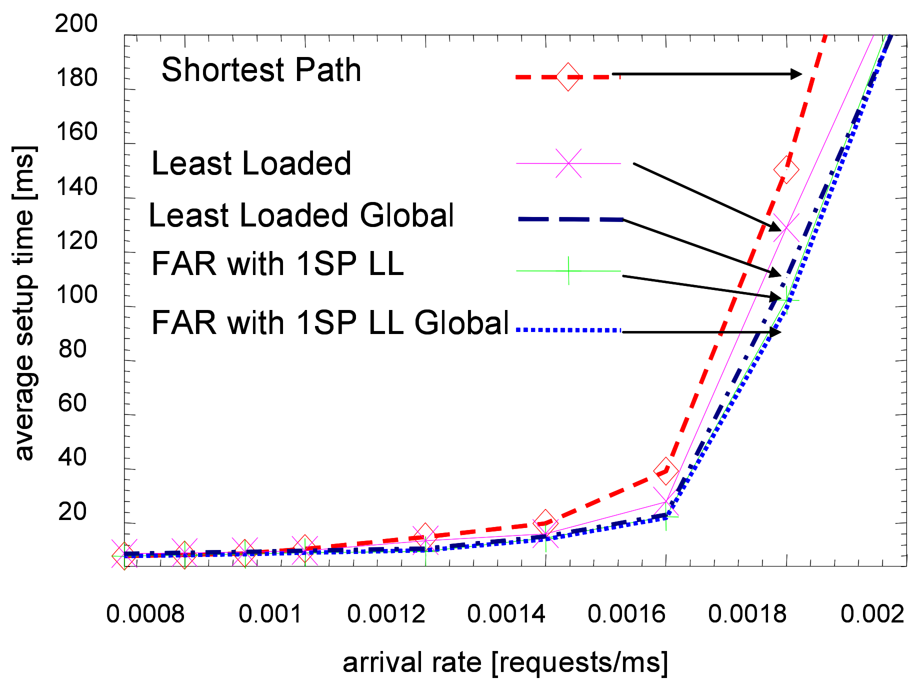


Figure 14: Average path setup time for Japan backbone network with dynamic link use information: $W = 8$, $1/\mu$ = 1.0ms

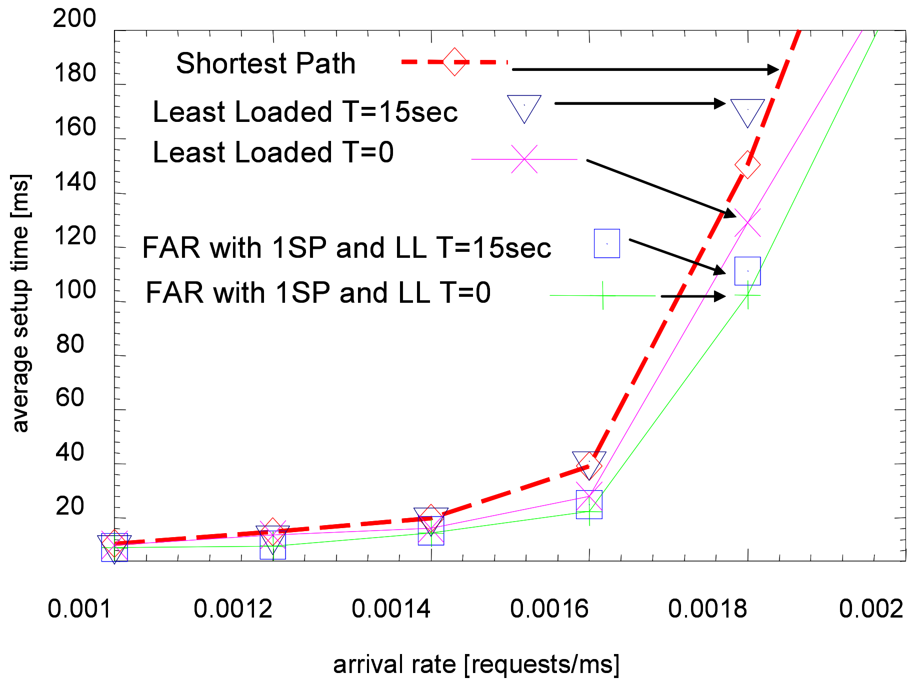


Figure 15: Average path setup time for Japan backbone network with periodic link use information: $W = 8$, $1/\mu$ = 1.0ms.

4 Implementation and experiments

The simulation results included some assumptions that did not reflect the actual network control plane well. We therefore implemented the lightpath network within the GMPLS standard and compared the actual computation overheads and simulation results. The GMPLS with the RSVP–TE and OSPF–TE protocols was suitable for lightpath network implementation, which we based on the current standard obtained from Internet draft documents [17, 25–34] and RFCs [13–15, 24, 35–38]. We will first explain the background to implementation, its overview, and discuss the results obtained from evaluating performance in the experiments, in Subsection 4.1, Subsection 4.2, and in 4.3, respectively. We will then discuss the accommodation of frequent path request and link–state–information exchange for this experimental environment.

4.1 Implementation background

There are presently no lightpath network implementations on PCs. Therefore, we implemented a program which run lightpath network control plane. The GMPLS (Generalized Multiprotocols Label Switching) achieves a label switching network on multi–protocols. A label has information on link attributes that includes the switching capabilities of packets, fibers, wavelength lambdas, and TDM (Time Division Multiplexing) time slots. Using the assigned labels, the network nodes switch the input interface bound to a lambda to an output interface bound to a lambda. The processing layers for path establishment in this GMPLS architecture follow the description in Fig. 16. On receiving a path establishment request, the routing plane selects the proper route for the request. The resource reservation plane then signals the reservation process to neighboring nodes along the control channel.

The RSVP–TE protocol leads the route information to the RSVP path message, which goes to the control channel through within an explicit route. On receiving the RSVP path message, the intermediate node confirms the path message's label information on whether it has the capabilities for the specified link or not and if it is capable of resource reservation then forwards this to the next node. On receiving the path message, the receiver node confirms the path message and returns the RSVP reserve message to the sender node with reservation and switching the label of the interfaces among the routes. Whenever the label is not capable or the reservation fails, an RSVP NACK signal is transmitted to the sender node to notify that establishment has been blocked.

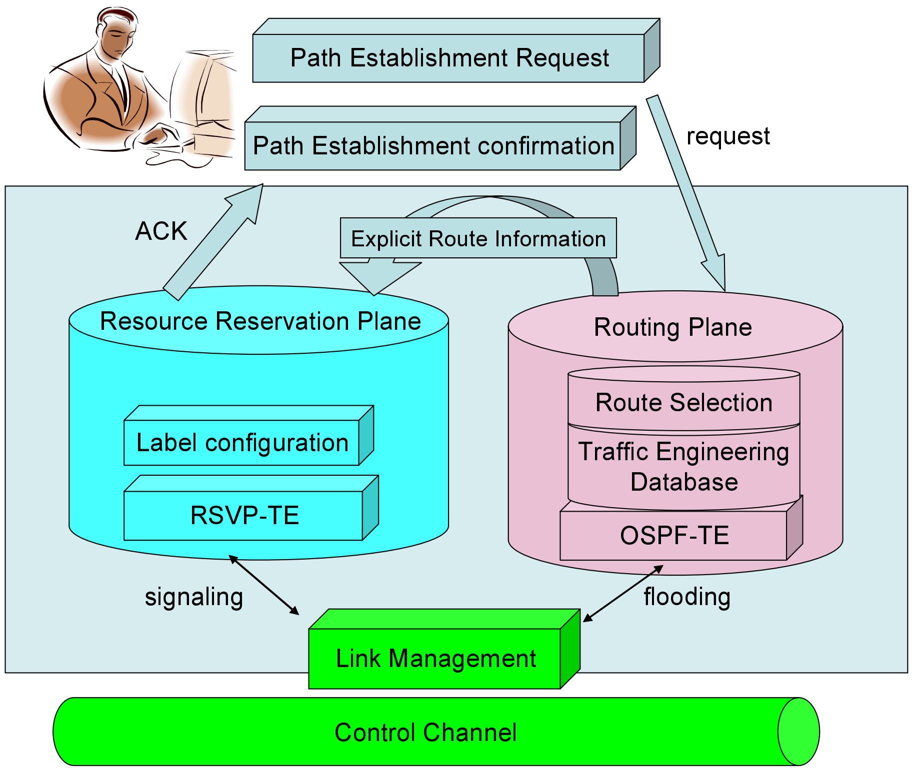


Figure 16: GMPLS signaling architecture and its control plane

The routing plane has the OSPF–TE signaling protocol, the Traffic Engineering database, and a routing algorithm. The OSPF–TE protocol floods information on attributes of links between neighboring nodes. The Traffic Engineering database collects the flooded OSPF data packets. Routing algorithm use the information from Traffic Engineering database, and select a suitable route.

Link Management manages the control channel, which provides fault discovery (e.g., link disconnection, node failure, and data transmission degradation). The control plane's signals are protected from any faults by the link management protocol.

4.2 Implementation overview

We implemented a lightpath network system on PCs running kernel 2.6.9. with C++ source code. We implemented the signaling protocols in the UDP transmission layer.

Figure 17 shows the modules we implemented. The control planes and our implemented modules correspond in terms of the resource reservation plane to the RSVP Controller and the routing plane to the OSPF Controller, and in terms of the Routing Algorithm and Resource Manager, and Link Management to the Link Management Protocol Controller. The Console Interface works the configuration command interpreter for administration for monitoring and inputting by the operator. The GMPLS System Manager receives messages from the Console Interface and commands the other signaling and controller modules. The Path Request Generator generates requests for path establishment and path release to the RSVP Controller. The frequency of requests is configured by the GMPLS System Manager. The RSVP Controller has a Request Queue, which holds excessive requests from the Path Request Generator. The RSVP Controller runs Path Establishment, Release, and RSVP signaling. When the RSVP Controller starts path establishment, it uses the route from the Routing Algorithm. When RSVP messages are received from the control plane, the RSVP Controller receives them and reserves or releases the designated wavelength commanding the Resource Manager. The RSVP signals are transmitted to the control plane through the Network Interface. The Network Interface has a Link Management Protocol Controller, which manages transmission errors and neighbor node notifications. The OSPF Controller manages link state information exchange, and when wavelength resource reservations changes, it floods the information to neighboring nodes through the control plane. The OSPF Controller also receives information flooded by neighboring nodes, and stores this in the Database. The information is used by the Routing Algorithm, which selects suitable route for the RSVP Controller. If the routing algorithm has fixed routing, the operator inputs the route for each destination node to the Console Interface, and the inputted route is received by the Routing Algorithm through the GMPLS Manager. To implement the GMPLS system, we have to monitor the processing time for each module, such as link–state exchange and route selection. The Log Manager collects the processing logs in a file from which the operator can confirm and validate performance evaluations in experiments.

4.3 Implementation experiments and validate the simulation

Figure 18 outlines the experimental environment. We used four Linux PCs that had an Intel® Pentium®4 3.00 Ghz CPU, 1GByte of DDR–RAM, and a gigabit–ethernet network interface. One network node consisted of one PC, which ran the implemented program. One link consisted of a 1000Base–TX ethernet cable, which was connected via a Dell® Power Connect 5224® switching hub.

In Section 3, we assumed that the calculation times for link–state–information exchange and

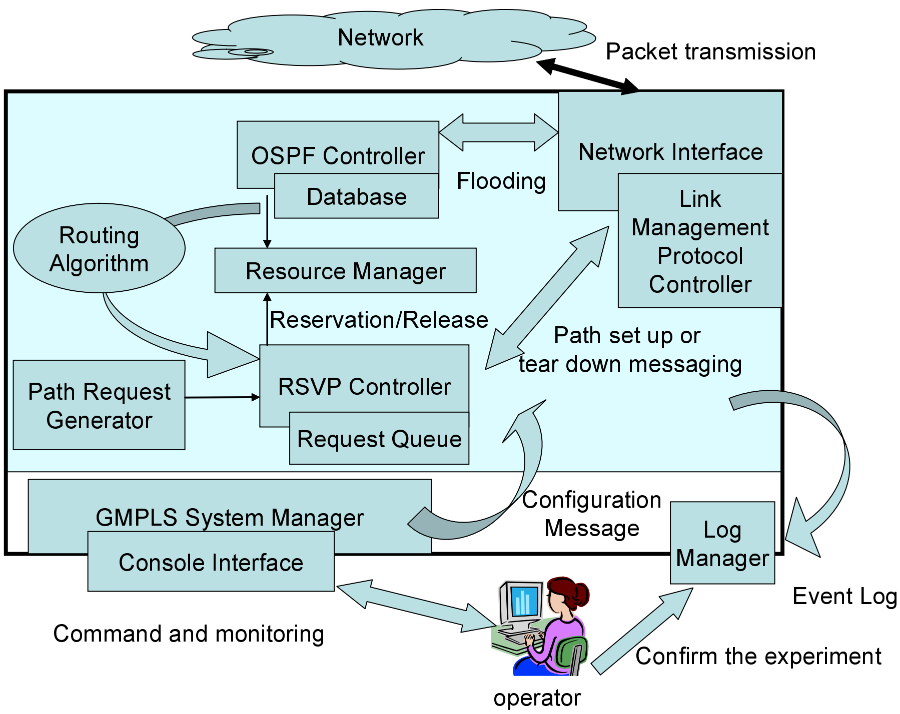


Figure 17: GMPLS implementation system overview

route selection would not cause CPU over load. In our implementation, these overheads were monitored (see Table 4) on a two–node and one– link network topology. As we can see from the results, our assumptions of regarding calculation time did not dominate the performance evaluations.

Figure 19 plots the results obtained from simulations and experiments. The average values of three experiments using different random seeds are plotted in Fig. 19. In each experiments, we run fifteen minutes, and we use the results of lightpath setup time at the last six minutes. From this

Table 4: Evaluation of calculation time: link–state advertisement and route selection

Measurements	Calculation Time [μsec]
Link State Advertisement	20
Fixed route selection	10
Least loaded route selection	15
Alternate route selection	10 and the other 14

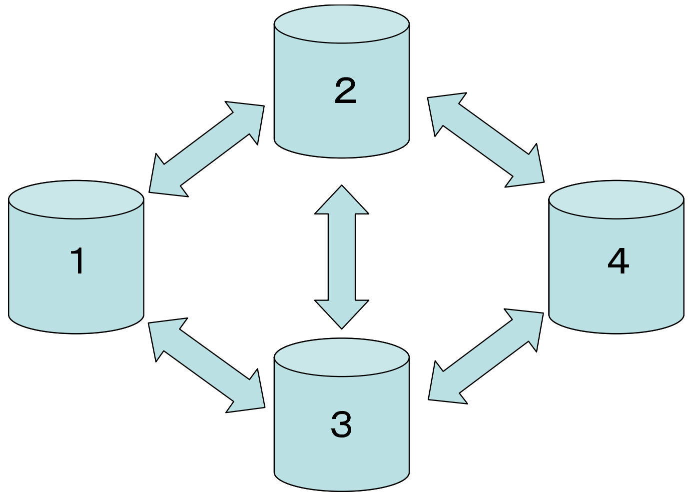
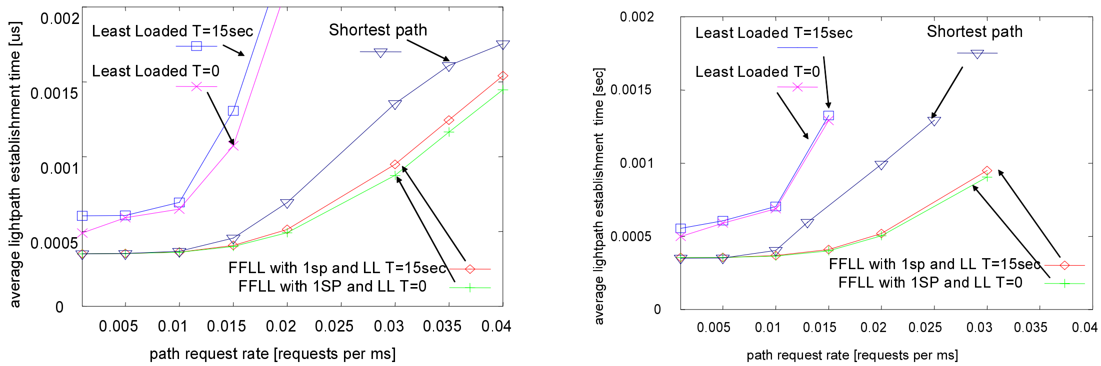


Figure 18: Network topology for implementation experiment

figure, we observe that the least-loaded routing algorithm performs worse than the others. This is because that almost all of the node pairs are directory connected in this topology, the advantage of the lowest amount of wavelength resources utilization combined with shortest path routing therefore become larger than that in the more larger network (as discussed in Section 3). We also see that the simulation results approximates the results obtained by experiments well at the lightly–loaded environment. However, in highly-loaded environment, the experiments did not work as the simulations. This is because the simulation did not reflect a buffer for handling control packets. In the actual network, each node has a socket buffer for handling control packets. When the buffer has insufficient capacities, lightpath requests are frequently rejected regardless of sufficient wavelength resources. Or nodes interpret wrongly that control channel is down, even if the control channel alives. As this experiment shows, the traffic on the control channel much affects the throughput of the system. We will therefore discuss the scalability of our implementations in Subsec 4.4.

4.4 Scalability of lightpath setup methods: routing overhead perspective

As we discussed in previous section, the control traffic for link state information exchange may degrade the system throughput. In this Subsection, we clarify this problem comparing some results obtained by experiments using several types of link–state–information triggers. We implemented three types of flooding–trigger method in OSPF controller; periodic, threshold–based,



(a) results obtained by Simulations (b) results obtained by Experiments

Figure 19: Evaluation of lightpath setup time comparing with simulation results and experiment results: $W = 8$, $1/\mu$ = 100ms

and full–information flooding. The periodic implementation is exactly the same as described in Section. 3.3. With the threshold-based flooding, each link is classified into three link state (low, middle, and high) according to the number of available wavelengths. The low state means that the link has lightly-loaded, i.e., has many available wavelengths. The high state means that the link has insufficient available wavelengths, whereas the middle state means the link has sufficient available wavelengths. In this experiment, we set low state to be 0-20%, middle state to be 20-60%, and high state to be 60-100%. OSPF controller observe the state of neighboring link. If the number of available wavelengths changes the state, OSPF controller floods link-state information. With the full-information flooding, the link-state information includes the availability of each wavelength. That is, each node can know which wavelength is currently used or un-used. Note that the backward reservation protocol does not need this information, while the forward reservation protocol does need. Here, we intend to reveal the disadvantages of the forward reservation protocol from the control traffic perspective.

Table 5 summarizes the arrival rate at which the lightpath setup time saturated dependent on the policies for link–state–information exchange. Note that the arrival rate is normalized by frequent link–state–information exchange. We can see that the threshold–based exchange performs better than the other exchange methods does. This fact can be explained as follows. At the saturation point, almost all of the wavelength resources are utilized. With frequent link–state–information

Table 5: Evaluation of traffic overhead: How rate of arrival make the system saturate: $W = 8$, $1/\mu = 100ms$

link–state–information exchange method	relative arrival rate
threshold based exchange	1.12
periodic exchange	1.10
frequent exchange	1.00
full–information frequent exchange	1.00

exchange method, there are huge number of control packets for link-state update, which lengthen the processing delay. The release operations for lightpaths are delayed, which in turn induces frequent retries of lightpath establishment and therefore the saturated arrival rate becomes lower. The threshold–based exchange method exchange the information only when link state changes among the three states, and thus the method show a good result at the saturated point.

4.5 Scalability of lightpath setup methods: signaling overhead perspective

Challenges for development of WDM technology increases the number of wavelengths on a fiber. By using a silica–based optical fibers, multiplexing of a thousand wavelengths on the fiber must be feasible in the future [39]. However, as the number of wavelengths on a fiber increases, the processing of control packets will eventually becomes a bottleneck since the backward reservation method requires *and*–operations for each wavelengths. In this subsection, we investigate the scalability of control–plane with regard to the number of wavelengths. We take the same experiments of Table 5 using the thousand wavelengths, and the results show (, but not presented in this thesis) that there are no difference on the saturation points among the exchange methods. Furthermore, at the saturation point, the maximum link utilization is around 40 %. In this case, there are little retries of lightpath establishments, thus, it is unlikely that the routing overheads make the system saturated. Table 6 shows the percentage of processing for experiments at the saturation point, which indicate that the PROBE packet processing actually becomes the bottleneck of the system. One possible approach to resolve this bottleneck is to introduce a hardware, such as ASIC or FPGA for handling the PROBE packet. Another approach is to introduce an optical code based processing [40].

Table 6: Evaluation of processing overheads: processing ratio at saturation points with frequent link–state–information exchange

	processing ratio[%]	
Types of processing	$W = 8$	$W = 1000$
PROBE handling	0.02	21.0
Reservation handling	0.01	0.50
Release handling	0.01	0.85
link–state–information exchange	58.2	20.1
route selection	10	0.50

In summary, protocol processing at a node will actually be a bottleneck for future optical networks. However, in the distributed wavelength–routed networks, the bottleneck goes to the PROBE packet processing, and thus introducing a hardware that handle the PROBE packet will greatly improve the performance of optical networks.

5 Conclusion

We investigated what effect delayed link–state information exchange had on reservation protocols. We evaluated them with three network topologies, i.e., a random mesh network, realistic mesh network, and three–node tandem network in computer simulations. The results revealed that when the backward reservation protocol was used, routing could be done with less frequent link–state information exchange using less detailed information than when the forward reservation protocol was used. We then evaluated alternate routing algorithms in terms of routing and delayed link–state information. The results revealed that when the primary route was fixed to the shortest path and the other routes were less loaded it performed with the save average lightpath establishment time as the least loaded algorithm with global link–state information. Furthermore, as the topology grew larger, the performance of the least loaded algorithm degraded because of imprecise link–state information, while that of the alternate routing algorithms did not. We also found the simulation results were valid in an actual network environment. And the results obtained by experiments show that frequent link–state–information exchange degrade the control channel. That is, the link–state–information should be flooded periodic, the backward reservation protocol and the alternate routing algorithms are good for distributed lightpath networks.

We implemented a control plane for lightpath network. Then we clarified the validation of the simulation results and the evaluation of scalability of the number of wavelengths with several types of link–state–information exchange methods. The results revealed that the threshold–based link–state-information exchange is good approach to reduce the consumption of bandwidths in control channel and lessen the link–state information exchange processing. As a result, it has more system throughput than other methods do where the number of wavelength is small. In a large number of wavelength environment, we also showed that processing for PROBE packets much degrade the system throughput.

In future studies, we intend to evaluate a lightpath network that provides for data communication. We intend to extend the functions of our implementation source codes to establish an optical data communication network with OXCs.

Acknowledgement

I am very grateful for the advice and support of my advisor, Professor Masayuki Muarata, for his feedback and for keeping me focussed in my research. My work have continued to be improved by his graceful advice.

All of my works can't be achieved without the supports of Research Assistant Shin'ichi Arakawa. I sincerely appreciate for him to greatly improve my work, and help preparations of my thesis.

Thank you to NiCT Hiroaki Harai for his advice which clarified this work. I am particularly thankful for the continued input of Associate Professor Naoki Wakamiya, Associate Professor Go Hasegawa, Research Assosiate Ichinoshin Maki, and Research Assosiate Masahiro Sasabe.

Finally, I owe a lot to my friend, colleague Yousuke Kanitani and Dr. Kenji Leipnitz, and Advanced Network Architecture Research Group of Osaka University. Our conversations and work together have greatly influenced this thesis.

References

[1] J. Zhou and X. Yuan, "A study of dynamic routing and wavelength assignment with imprecise network state information," in *Proceedings of International Conference on Parallel Processing Workshops (ICPPW'02)*, pp. 202–207, Aug. 2002.

[2] H. Zang, J. Jue, L. Sahasrabuddhe, R. Ramamurthy, and B. Mukherjee, "Dynamic lightpath establishment in wavelength–routed WDM networks," *IEEE Communication Magazine*, vol. 39, pp. 100–108, Sept. 2001.

[3] H. Alnuweiri, L.-Y. K. Wong, and T. Al-Khasib, "Performance of new link state advertisement mechanisms in routing protocols with traffic engineering extensions," *IEEE Communications Magazine*, pp. 151–162, May 2004.

[4] J. Li, G. Mohan, E. C. Tien, and K.-C. Chua, "Dynamic routing with inaccurate link state information in integrated IP over WDM networks," *Computer Networks*, vol. 46, pp. 829–851, Dec. 2004.

[5] H. Liu, E. Bouillet, D. Pendarakis, N. Komaee, and J.-F. Labourdette, "Distributed route communication and provisioning in shared mesh optical networks," *IEEE Journal on Selected Areas in Communications*, vol. 22, pp. 1626–1639, Nov. 2004.

[6] X. Yuan, R. Melham, and R. Gupta, "Distributed path reservation for multiplexed all-optical networks," *IEEE Transactions on Computers*, vol. 48, pp. 1355–1363, Dec. 1999.

[7] B. Ganguly and E. Modiano, "Distributed algorithms and architectures for optical flow switching in WDM networks," in *Proceedings of IEEE Symposiun on Computers and Communications (ISCC'2000)*, pp. 134–139, July 2000.

[8] E. Karasan and E. Ayanoglu, "Effects of wavelength routing and selection algorithms on wavelength conversion gain in WDM optical networks," *IEEE/ACM Transaction on Networking*, vol. 6, pp. 186–196, Apr. 1998.

[9] S. Sartzetakis, C. Tziouvaras, and L. Georgiadis, "Adaptive routing algorithm for lambda switching networks," *Optical Networks Magazine*, vol. 3, pp. 476–487, July 2002.

[10] J. Strand, A. Chiu, and R. Tkach, "Issues for routing in the optical layer," *IEEE Communication Magazine*, vol. 39, pp. 81–87, Feb. 2001.

[11] H. Zang, J. Jue, and B. Mukherjee, "A review of routing and wavelength assignment approaches for wavelength–routed optical WDM networks," *Optical Networks Magazine*, vol. 1, pp. 47–60, Jan. 2000.

[12] J. P. Jue and G. Xiao, "An adaptive routing algorithm for wavelength–routed optical networks with a distributed control scheme," in *Proceedings of Ninth International Conference on Computer Communication and Networks (IC3N'2000)*, pp. 192–197, Oct. 2000.

[13] E. Mannie, "Generalized multi–protocol label switching (GMPLS) architecture," Request for Comments (RFC) 3945, Internet Engineering Task Force, Oct. 2004. `http://www.ietf.org/rfc/rfc3945.txt`.

[14] D. Katz, K. Komplla, and D. Yeung, "Traffic engineering (TE) extensions to OSPF version 2," Request for Comments (RFC) 3630, Internet Engineering Task Force, Sept. 2003. `http://www.ietf.org/rfc/rfc3630.txt`.

[15] L. Berger, "Generalized Multi-Protocol Label Switching (GMPLS) Signaling Resource ReSerVation Protocol-Traffic Engineering (RSVP-TE) Extensions," Request for Comments (RFC) 3473, Internet Engineering Task Force, Jan. 2003. `http://www.ietf.org/rfc/rfc3473.txt`.

[16] D. Papadimitriou, F. Poppe, and B. Rousseau, "Application of the link management protocol to discovery and forwarding adjacencies," *Photonic Network Communications*, vol. 8, pp. 23–37, June 2004.

[17] J. Lang, "Link management protocol (LMP)," Internet–Draft Version 10, Internet Engineering Task Force, Oct. 2003. available at `http://www.ietf.org/internet-drafts/draft-ietf-ccamp-lmp-10.txt`.

[18] S. Subramaniam and R. Barry, "Wavelength assignment in fixed routing WDM networks," in *Proceedings of IEEE International Conference on Communications (ICC'97)*, vol. 1, pp. 406–410, June 1997.

[19] A. Mokhtar and M. Azizoglu, "Adaptive wavelength routing in all–optical networks," *IEEE/ACM Transactions on Networking*, vol. 9, pp. 553–566, Oct. 2001.

[20] R. Ramamurthy and B. Mukherjee, "Fixed–alternate routing and wavelength conversion in wavelength–routed optical networks," *IEEE/ACM Transactions on Networking (TON)*, vol. 10, pp. 351–367, June 2002.

[21] P.-H. Ho and H. T. Mouftah, "A novel distributed protocol for path selection in dynamic wavelength–routed WDM networks," *Photonic Network Communications*, vol. 5, pp. 23–32, Jan. 2003.

[22] H. Harai, M. Murata, and H. Miyahara, "Performance of alternate routing methods in all–optical switching networks," in *proceedings of INFOCOM'97*, vol. 2, pp. 416–421, Apr. 1997.

[23] D. Eppstein, "Finding the k shortest paths," in *Proceedings of IEEE Symposium on Foundations of Computer Science*, pp. 154–165, Nov. 1994.

[24] J. Moy, "OSPF version 2," Request for Comments (RFC) 1247, Internet Engineering Task Force, July 1991. `http://www.ietf.org/rfc/rfc1247.txt`.

[25] D. Papadimitriou, J. Drake, J. Ash, A. Farrel, and L. Ong, "Requirements for generalized mpls (GMPLS) signaling usage and extensions for automatically switched optical network (ASON)," Internet–Draft Version 07, Internet Engineering Task Force, Oct. 2004. available at `http://www.ietf.org/internet-drafts/draft-ietf-ccamp-gmpls-ason-reqts-0%7.txt`.

[26] L. Berger, "GMPLS signaling procedure for egress control," Internet–Draft Version 03, Internet Engineering Task Force, Aug. 2004. available at `http://www.ietf.org/internet-drafts/draft-ietf-ccamp-gmpls-egress-contr%ol-03.txt`.

[27] K. Kompella and Y. Rekchter, "Routing extensions in support of generalized multi–protocol label switching," Internet–Draft Version 09, Internet Engineering Task Force, Oct. 2003. available at `http://www.ietf.org/internet-drafts/draft-ietf-ccamp-gmpls-routing-09.t%xt`.

[28] G. Swallow, J. Drake, and H. Ishimatsu, "Generalize multiprotocol label switching (GMPLS) user–network intertface (UNI): Resource reservation protocol–traffic engineering (RSVP–TE) support for the overlay model," Internet–Draft Version 05, Internet Engineering Task Force, Oct. 2004. available at `http://www.ietf.org/internet-drafts/draft-ietf-ccamp-gmpls-overlay-05.t%xt`.

[29] K. Kompella and Y. Rekhter, "OSPF extensions in support of generalized multi–protocol label switching," Internet–Draft Version 12, Internet Engineering Task Force, Oct. 2003. available at `http://www.ietf.org/internet-drafts/draft-ietf-ccamp-ospf-gmpls-extensi%ons-12.txt`.

[30] K. Shimano, K. Shiomoto, and Yoshihiki, "Extra class LSP service using protecting resources in GMPLS networks recovery (protection and restoration) terminology," Internet–Draft Version 02, Internet Engineering Task Force, Feb. 2004. available at `http://www.pilab.org/draft/downloads/draft-pil-ccamp-extra-lsp-02.txt`.

[31] A. Fredette and J. Lang, "Link management protocol (LMP) for dense wavelength division multiplexing (DWDM) optical line systems," Internet–Draft Version 03, Internet Engineering Task Force, Dec. 2003. available at `http://www.ietf.org/internet-drafts/draft-ietf-ccamp-lmp-wdm-03.txt`.

[32] Z. Ali, D. Praire, and D. Papadimitriou, "Node ID based RSVP hello: A clarification statement," Internet–Draft Version 01, Internet Engineering Task Force, Oct. 2004. available at `http://www.ietf.org/internet-drafts/draft-ietf-ccamp-rsvp-node-id-based%-hello-01.txt`.

[33] M. Dubuc, S. Dharanikota, T. Nadeau, J. Lang, and E. McGinnis, "Link management protocol management information base," Internet–Draft Version 10, Internet Engineering Task Force, Sept. 2004. available at `http://www.ietf.org/internet-drafts/draft-ietf-ccamp-lmp-mib-10.txt`.

[34] J. Drake, D. Papadimitriou, A. Farrel, D. Brungard, Z. Ali, A. Ayyangar, H. Ould-Brahim, and D. Fedyk, "Generalized mpls (GMPLS) RSVP–TE signaling in support of automatically switched optical network (ASON)," Internet–Draft Version 02, Internet Engineering

Task Force, July 2004. available at `http://www.ietf.org/internet-drafts/draft-ietf-ccamp-gmpls-rsvp-te-ason%-02.txt`.

[35] R. Coltun, D. Ferguson, and J. moy, "OSPF for IPv6," Request for Comments (RFC) 2740, Internet Engineering Task Force, Dec. 1999. `http://www.ietf.org/rfc/rfc2740.txt`.

[36] L. Berger, "Generalized multi-protocol label switching (gmpls) signaling functional description," Request for Comments (RFC) 3471, Internet Engineering Task Force, Jan. 2003. `http://www.ietf.org/rfc/rfc3471.txt`.

[37] P. Ashwood-Smith, "Generalized multi–protocol label switching (GMPLS) signaling constraint-based routed label distribution protocol (CR–LDP) extensions," Request for Comments (RFC) 3472, Internet Engineering Task Force, Jan. 2003. `http://www.ietf.org/rfc/rfc3472.txt`.

[38] K. Kompella and Y. Rekhter, "Signalling unnumbered links in resource reservation protocol - traffic engineering (RSVP-TE)," Request for Comments (RFC) 3477, Internet Engineering Task Force, Jan. 2003. `http://www.ietf.org/rfc/rfc3477.txt`.

[39] Masayuki Muarata and Ken'ichi Kitayama and Masaharu Ohashi and Hideo Miyahara, "IP over a–thousand–wavelength division multiplexing: Is it possible for resolving the network bottlenecks?," *IEICE Technical Report* (PNI2000-6), pp. 41–48, May 2000.

[40] Shin'ichi Arakawa and Masayuki Murata and Ken'ichi Kitayama, "One-way reservation scheme using optical code processing for fast data transfer in WDM networks," in *Proceedings of The 7th IFIP Working Conference on Optical Network Design and Modeling (ONDM 2003)*, pp. 1073–1088, Feb. 2003.