Reconfiguration of Logical Topologies with Minimum Traffic Disruptions in Reliable WDM-Based Mesh Networks

Shinya Ishida (e-mail: s-isida@ist.osaka-u.ac.jp) *
Graduate School of Information Science and Technology, Osaka University
Machikaneyama 1-32, Toyonaka, Osaka, 560-0043 Japan

Shin'ichi Arakawa (e-mail: arakawa@econ.osaka-u.ac.jp)
Graduate School of Economics, Osaka University
Machikaneyama 1-7, Toyonaka, Osaka, 560-0043 Japan

Masayuki Murata (e-mail: murata@cmc.osaka-u.ac.jp)
Cybermedia Center, Osaka University
Machikaneyama 1-32, Toyonaka, Osaka, 560-0043 Japan

Abstract

A Wavelength Division Multiplexing (WDM) network offers a flexible networking infrastructure by assigning the route and wavelength of lightpaths. We can construct an optimal logical topology, by properly setting up the lightpaths. Furthermore, setting up a backup lightpath for each lightpath improves network reliability. When traffic demand changes, a new optimal (or sub-optimal) topology should be obtained by again applying the formulation. Then, we can reconfigure the running topology to the logical topology obtained. However, during this reconfiguration, traffic loss may occur due to the deletion of older lightpaths. In this paper, we consider reconfiguring the logical topology in reliable WDM-based mesh networks, and we propose five procedures that can be used to reconfigure a running lightpath to a new one. Applying the procedures one by one produces a new logical topology. The procedures mainly focus on utilizing free wavelength resources and the resources of backup lightpaths, which are not used usually for transporting traffic. The results of computer simulations indicate that the traffic loss is remarkably reduced in the 14-node network we used as an example.

keyword: WDM (Wavelength Division Multiplexing), lightpath, logical topology, reconfiguration, reliability

---

* Send correspondence to Shinya Ishida (e-mail: s-isida@ist.osaka-u.ac.jp, tel: +81-6-6850-6863, fax: +81-6-6850-6868).

1 Introduction

Recently, the rapid growth of Internet traffic has led to demands for extra capacity in the backbone networks. WDM (Wavelength Division Multiplexing) technology, which allows multiple wavelengths to be carried on a single fiber, is expected to handle those demands with lower costs. Furthermore, recent advances in optical devices, such as optical switches, have led to WDM technology with networking capabilities. Suppose that each node has an optical switch directly connecting each input wavelength to an output wavelength, so that there is no electronic processing at the packet level. That is, no electronic routing is needed at the nodes. This means a wavelength path can be set up directly between two nodes via one or more optical switches.

Logical topologies are constructed as sets of lightpaths, each of which being configured by properly assigning a wavelength and determining a route. In Refs. [1, 2], methods to design logical topologies are investigated. In many past studies on designing a logical topology, it has commonly been assumed that traffic demand is known in advance, except [3, 4]. However, in practice, it is difficult to predict changes in traffic demand due to factors like the start of new services in networks, such as streaming media services and content delivery services. Required lightpaths should be increased immediately once the WDM network performance becomes inadequate. Therefore, flexible network design is a more important method than static network design [5].

As the transmission capacities of WDM networks increase, traffic losses when failures occur in the networks are unacceptable. Protection methods are considered to improve the reliability of WDM networks, in which each lightpath has wavelength resources reserved for backup purpose [6]. In this method, a primary and its backup lightpath are assigned for each traffic demand. The *primary lightpath* is used for usual data transmission. *Backup lightpath* is used when the primary lightpath is unavailable due to a network failure. An advantage of the protection method is to avoid computing a route and a wavelength of a backup lightpath after the failure occurs, and thus fast recovery can be achieved. However, the total capacity that a WDM network accommodate is also limited, since the wavelength resources of backup lightpaths are not used for traffic transportation. To avoid the waste of wavelength resources for backup lightpaths, the *shared protection strategy* is proposed [7]. Using this strategy, several backup lightpaths can share wavelength resources if and only if any of the backup lightpaths are not utilized simultaneously.

In Ref. [7], incremental capacity dimensioning is proposed. It is a network design method to achieve flexibility and reliability in WDM mesh networks. In this method, logical topology design is first applied for a given traffic demand. With the incremental traffic demand, a one-by-one assignment of the primary lightpath as well as the backup lightpaths based on the user's perception of performance is done. The one-by-one assignment may result in a topology far from the optimal one. Therefore, we reconfigure the running topology to the optimal (or sub-optimal) logical topology obtained by again applying the design method for the current traffic demand.

During reconfiguration between two logical topologies, packet loss or delayed arrival may occur due to the deletion of older lightpaths. It may result in a loss of traffic on those lightpaths and decline of the performance of a network. Therefore, there is a trade-off in the reconfiguration between improved network performance obtained from the reconfiguration itself, and the traffic loss penalty due to the deletion of the lightpath during the reconfiguration [8]. There have been various studies on reconfiguration methods to minimize lightpath deletion [8–12]. In Ref. [11], a branch-exchange method is proposed to relieve the influence of tearing down working lightpaths. It tries to minimize traffic loss by reducing the number of steps required in a reconfiguration. Reconfigurations in ring networks are also considered in Ref. [12]. However, most of the studies on the reconfiguration have been proposals for star-based WDM networks with optical passive star couplers or ring-based WDM networks. In this paper, on the other hand, we propose a reconfiguration algorithm for WDM-based mesh networks to provide flexible and reliable backbones. Our basic idea for the reconfiguration is to use wavelength resources reserved for backup lightpaths which are not always utilized. Our algorithm is based on five procedures to set up and tear down lightpaths. In addition to simply setting up or tearing down lightpaths, we have considered three other procedures to incorporate wavelength resources for backup lightpaths. Since the backup lightpaths are not always used for transporting the actual traffic, we exploit their wavelength resources assuming that failure does not occur during reconfiguration. We evaluated the performance of the algorithm with a 14-node network. The evaluation results show that the algorithm can dramatically reduce the incidence of traffic loss during reconfigurations against random changes of traffic patterns. In certain situations, we achieved reconfigurations without traffic loss.

This paper is organized as follows. In Section 2, we introduce five procedures to reconfigure local topologies. In Section 3, we present a heuristic reconfiguration algorithm. We evaluate the algorithm in Subsection 4.1 with random traffic models. We also evaluate the performance of the algorithm with taking traffic transition and the rate of utilized resources into account in Subsection 4.2. We conclude our paper in Section 5.

2 Procedures to reconfigure logical topologies

In this section, we introduce five procedures to reconfigure logical topologies: SWITCH, APPEND, BACKUP, RELEASE, and DELETE. Here, we define a *working lightpath* as a lightpath on a current working logical topology on which data traffic is actually transported. We also define a *target lightpath* as a new lightpath organizing a part of the new logical topology obtained by a certain logical topology design algorithm. In followings, the shared protection method is assumed, but these procedures are easily applied for the dedicated protection.

2.1 Notations

First, let us explain the symbols used in our algorithm.

N: Number of nodes in a network. Each node is assigned a number from 1 to N, respectively.

W: Degree of wavelength multiplexing. Each wavelength is assigned an index number from 1 to W, respectively.

L_1: Set of working lightpaths included in a current logical topology.

L_2: Set of target lightpaths included in a target logical topology.

$b(l)$: Backup lightpath of a lightpath l.

$s(l)$: Source node of a lightpath l.

$d(l)$: Destination node of a lightpath l.

$\lambda(l)$: Wavelength allocated to a lightpath l. $1 \leq \lambda(l) \leq W$.

2.2 SWITCH procedure

Traffic loss is one of the fatal problems during the reconfiguration of logical topologies. A reconfiguration of a logical topology has to be implemented rapidly and smoothly even though there may be a significant traffic flow through the logical topology. If a working lightpath is deleted carelessly, the traffic on it is of course lost and the network performance gets worsen. However, the SWITCH procedure can reduce such a traffic loss remarkably by switching traffic from a current lightpath l_1 to a target lightpath l_2. These two lightpaths have the same source and destination nodes. The SWITCH procedure is as follows:

Step 1: Reserve wavelength resources for a target lightpath l_2, where $s(l_1) = s(l_2)$ and $d(l_1) = d(l_2)$ are identical, i.e., the source and destination nodes of the working lightpath are identical to the source and destination nodes of a target lightpath. If the resource reservation is succeeded, go to Step 2. Otherwise quit this procedure.

Step 2: Set the target lightpath l_2.

Step 3: Switch the traffic on the working lightpath l_1 to the target l_2.

Step 4: When the last packet on the working lightpath reaches the destination node $d(l_1)$, go to Step 5.

Step 5: Delete the working lightpath l_1 and its backup lightpath $b(l_1)$.

Figure 1 details the SWITCH procedure. If a portion of the wavelength resources utilized in a working lightpath are required to set up a target lightpath, the working lightpath is to be deleted. Here, we search a target lightpath which has the same source and destination nodes as the working lightpath. The traffic on the working lightpath is switched to the target lightpath before the former is deleted. Thus, traffic loss does not occur. This procedure progresses the reconfiguration effectively because a target lightpath is set up and a working lightpath is released without traffic loss. Hereafter, we describe $SWITCH(l_1, l_2)$ as the SWITCH procedure call for a working lightpath l_1 and a target lightpath l_2.

The backup lightpath of the target lightpath l_2 (i.e., $b(l_2)$) is not prepared in this SWITCH procedure because we do not consider failure occurrences during reconfiguration. The backup lightpaths are prepared after setting up the target lightpaths (see

Sec 3). If a failure is detected during a reconfiguration, restoration techniques may be applied.

2.3 APPEND procedure

The SWITCH procedure has the hard constraint that a working and target lightpath must have the same source and destination nodes to apply the procedure between these two lightpaths. If there is a target lightpath which cannot be set with the SWITCH procedure, the target lightpath is to be set with the following APPEND procedure. This procedure simply set the target lightpath. We note that the SWITCH procedure is more efficient than the APPEND procedure because the former releases the wavelength resources reserved for a working and its backup lightpath without traffic loss. $APPEND(l_2)$ works as follows to create a target lightpath l_2.

Step 1: Reserve wavelength resources for a target lightpath l_2. If the reservation is succeeded, go to Step 2. Otherwise, quit this procedure.

Step 2: Set the target lightpath l_2.

2.4 BACKUP procedure

If a portion of the wavelength resources for a working lightpath is required by one or more target lightpaths, and if there are no target lightpaths whose source and destination nodes are identical to those of the working lightpath, the working lightpath will be discarded without protecting its traffic. This usually results in traffic loss, but in a reliable WDM network with backup lightpaths, it is possible to avoid this loss by utilizing the backup lightpath prepared for the working lightpath (BACKUP procedure). The BACKUP procedure is as follows:

Step 1: Reserve wavelength resources for the backup lightpath $b(l_1)$ of a working primary lightpath l_1.

Step 2: Set the backup lightpath $b(l_1)$.

Step 3: Switch the traffic on the working lightpath l_1 to its backup $b(l_1)$.

Step 4: When the last packet on the working lightpath reaches the destination node $d(l_1)$, go to Step 5.

Step 5: Delete the working lightpath l_1.

Figure 2 illustrates the BACKUP procedure. Here, the working and its backup lightpaths are prepared from nodes 1 to 4. Suppose that the working lightpath is not necessary in the target logical topology and that there is not a target lightpath pair for the SWITCH procedure. In this situation, the working lightpath will be finally deleted. However, if the backup lightpath of the working lightpath is available, i.e., if all the wavelength resources of the backup lightpath are not shared with other backup lightpaths and are not required to set up target lightpaths, the traffic running through the working lightpath can be switched onto the backup lightpath. The backup lightpath is left until the reconfiguration of all the target primary lightpaths has been finished. We describe $BACKUP(l_1)$ as the procedure call to switch the traffic on a lightpath l_1 to its backup lightpath $b(l_1)$.

2.5 RELEASE procedure

Suppose that there is a working lightpath whose resources are required to set up target lightpaths and that the SWITCH, APPEND or BACKUP procedure cannot be applied. In this case, the traffic of the working lightpath tends to be lost by deleting it. However, with the RELEASE procedure, we can avoid the unnecessary deletions of working lightpaths.

The RELEASE procedure releases the wavelength resources of the backup lightpath for a working lightpath. After that, the reconfiguration continues to make progress using the SWITCH or APPEND procedure with the released wavelength resources. $RELEASE(l_1)$ is the procedure call to release the wavelength resources of a backup lightpath $b(l_1)$;

Step 1: Release wavelength resources reserved for a backup lightpath.

2.6 DELETE procedure

The DELETE procedure deletes a working lightpath without protecting its traffic. The traffic on the deleted lightpath is lost. Note that the DELETE procedure is the only procedure that results in a traffic loss. The working lightpath should be applied the RELEASE procedure before applying the DELETE procedure. $DELETE(l_1)$ is the procedure call to tear down a lightpath l_1;

Step 1: Release wavelength resources reserved for a working lightpath.

2.7 Wavelength re-allocation

Reconfigurations of logical topologies can be achieved using those five procedures. However, there is plenty of room for improvement in making the reconfigurations more efficient. Wavelength re-allocation provides one means. It re-allocates a new wavelength to a target lightpath, if another wavelength resources on the same route of the target lightpath are available. This extension is applied to the SWITCH, APPEND, or BACKUP procedures. Suppose that the reserved wavelength resources on a wavelength (say λ_1) are required to set up a target lightpath. Wavelength re-allocation assigns another available wavelength (assume λ_i as $i \neq 1$) to the target lightpath, which solves the conflict on λ_1. After that, the SWITCH or APPEND procedure sets up the target lightpath that uses the new λ_i wavelength resources.

This re-allocation is done when a portion of the wavelength resources of the target lightpath are reserved for working or backup lightpaths, and when there are free wavelength resources that are not required to set up other target lightpaths. When we use the wavelength re-allocation, Step 1 of the SWITCH procedure is revised as follows:

Step 1.1: Reserve wavelength resources for a target lightpath l_2, where $s(l_1) = s(l_2)$ and $d(l_1) = d(l_2)$. If the resource reservation is succeeded, go to Step 2. Otherwise go to Step 1.2.

Step 1.2: For each wavelength w $(1 \leq w \leq W,\ w \neq \lambda(l_2))$, try Step 1.3. If all trials fail, quit this procedure.

Step 1.3: Check the resources of wavelength w along the route of the target path. If the resources are not reserved and not required to set up any other lightpaths, go to Step 1.4. Otherwise go back to Step 1.2.

Step 1.4: Re-allocate w to the target lightpath and reserve the re-allocated resources. Go to Step 2.

Here, we search a free wavelength by the First-Fit policy in this re-allocation. The network performance of the re-allocated lightpath is identical to the original because the routes of both lightpaths seen from the upper layer are the same. Therefore, this wavelength re-allocation does not have any bad effect on the reconfigurations of logical topologies.

3 Reconfiguration algorithm

To relieve the unbalance of the traffic load on WDM mesh networks, we need reconfigurations of logical topologies. In this section, we propose a reconfiguration algorithm of logical topologies in WDM mesh networks. This algorithm is composed of five procedures described above. Since only the DELETE procedure leads to traffic loss, we heuristically suppress the number of DELETE procedure calls during a reconfiguration. It requires two logical topologies: a current logical topology and a target logical topology to be reconfigured. Therefore, the increase or decrease of traffic volume during the reconfiguration is not considered here.

We use a variable P in our reconfiguration algorithm to store the number of SWITCH, APPEND, BACKUP procedure calls in each iteration. We also define C as a set of working lightpaths which are candidates for a pair of a target lightpath to execute the SWITCH procedure. In this paper, we assume that no network failures occur during reconfiguration, because reconfiguration is invoked once per, say, one week or month.

3.1 Flow of reconfiguration algorithm

The reconfiguration algorithm we propose is as follows:

Step 1: For each target lightpath $l_2 \in L_2$, if there is a working lightpath $l_1 \in L_1$ which has the same route and wavelength as l_2, delete the elements from L_1 and L_2. $P \leftarrow 0$. $C \leftarrow \phi$.

Step 2: For each target lightpath l_2, try following steps. After that, go to Step 3.

Step 2.1: Add working lightpaths $l_1 \in L_1$ into C which fulfills these conditions: $s(l_1) = s(l_2)$, $d(l_1) = d(l_2)$ and l_1 does not reserve wavelength resources required for l_2. If $C \neq \phi$, go to Step 2.2. Otherwise, go to Step 2.3.

Step 2.2: Among the elements in C, select a working lightpath $l_1^{'}$ whose wavelength resources of both primary and backup lightpaths are most utilized to set up target lightpaths in L_2. Execute $SWITCH(l_1^{'}, l_2)$. Delete $l_1^{'}$ and l_2 from L_1 and L_2, respectively. $P \leftarrow P+1$. $C \leftarrow \phi$. Go back to Step 2.

Step 2.3: Execute $APPEND(l_2)$. Delete l_2 from L_2. $P \leftarrow P+1$. Go to Step 2.

Step 3: If $L_2 = \phi$, go to Step 6. Otherwise, go to Step 4.

Step 4: For each working lightpath $l_1 \in L_1$, which meets that there are no target lightpaths $l_2 \in L_2$ such as $s(l_1) = s(l_2)$ and $d(l_1) = d(l_2)$, execute $BACKUP(l_1)$ and, if it succeeds, delete l_1 form L_1 and $P \leftarrow P+1$.

Step 5: If $P > 0$, $P \leftarrow 0$ and go back to Step 2. Otherwise, go to Step 5.1.

Step 5.1: If there are working lightpaths whose backup lightpaths are not released, go to Step 5.2. Otherwise, go to Step 5.3.

Step 5.2: Select a working lightpath l_1 where the wavelength resources of backup lightpath ($b(l_1)$) are most utilized to set up target lightpaths in L_2. Execute $RELEASE(l_1)$. $P \leftarrow 0$. Go back to Step 2.

Step 5.3: Select a working lightpath l_1 whose wavelength resources of primary lightpath are most utilized to set up target lightpaths in L_2. Execute $DELETE(l_1)$. Delete l_1 form L_1. $P \leftarrow 0$. Go back to Step 2.

Step 6: Delete all of the remaining working lightpaths in L_1 and those backup lightpaths.

Step 7: Restore the re-allocated wavelengths of the target lightpaths to the original wavelengths.

Step 8: Reserve the wavelength resources of the backup lightpaths for the target lightpaths.

In Step 1, we detect working lightpaths which are also included in L_2. These working lightpaths are left as target lightpaths. From Steps 2 through 5, we set all the target lightpaths. Backup lightpaths for each target lightpath are set in Step 8. This is based on the assumption that no network failures occur during a reconfiguration.

In Step 2, we check whether the wavelength resources to set target lightpaths are available or not. We give priority to the SWITCH procedure over the APPEND procedure because of the differences in their efficiency (see Subsection 2.3). Hence, we try to apply the SWITCH procedure in setting up the target lightpath at first (Step 2.2). In Step 2.2, a working lightpath, $l_1^{'}$, is chosen as a pair of a target lightpath, l_2, from the set of the candidates.

To make the SWITCH procedure more efficiently, we select the pair heuristically as follows. The wavelength resources released by the SWITCH procedure can be utilized for setting up remaining target lightpath. We therefore choose a working lightpath as a pair of a target lightpath, such that the working lightpath holds the most amount of wavelength resources required to set up other remaining target lightpaths. Other strategies are to select lightpaths in descending (ascending) order of the number of hop-counts. We call these strategies longest-first-strategy (shortest-first-strategy).

As the reconfiguration continues by the SWITCH or APPEND procedure, target lightpaths reserve their wavelength resources. And available wavelength resources are decreased. Ultimately, no target lightpaths can be set because the available wavelength resources are exhausted. In Step 3, if all target lightpaths in L_2 have already been created, we can go to Step 6. Otherwise, in Step 4, we try to find free wavelength resources to utilize set up other remaining target lightpaths by applying BACKUP procedures. If one or more trial succeeds, we obtain new available wavelength resources without traffic loss. Then we go back to Step 2 and try to set up the rest of the target lightpaths in L_2.

In Step 6, all target lightpaths are created and traffic in a network is accommodated by these target lightpaths. Hence, we can delete the old working lightpaths in L_1. If there are lightpaths whose wavelengths are re-allocated by wavelength re-allocation, we tune the re-allocated wavelength to the original one in Step 7. In Step 8, reserve the wavelength resources for the backup lightpaths of the target lightpaths and finish the reconfiguration.

In this paper, we assume that one by one operations of reconfiguration procedures. However, two operations may be executed at the same time as long as the corresponding lightpaths are independent of the operations. Doing this results in shorter time to complete the reconfiguration, but it is out of the scope of this paper.

The operations of these reconfigurations are controlled in both centralized or distributed systems.

3.2 Heuristic selection strategy

Among the five procedures described in Section 2, the SWITCH, RELEASE and DELETE procedures enables wavelength resources to be available. If there are several working lightpaths to which these procedures can be applied, we select one working lightpath (and then release or delete it) such that other procedures can be applied efficiently.

In this paper, we propose a heuristic selection strategy to select the working lightpath. Our strategy select the working lightpath which holds most conflicts with target lightpaths. By releasing or deleting it maximize the number of target lightpaths to be set up. This strategy is used in Step 2.2, Step 5.2 and Step 5.3. Figure 3 depicts an example of our strategy. Here, we assume that a target lightpath from node 1 to node 9 via node 4 is being set up by the SWITCH procedure and there exists three candidate working lightpaths ($1 \rightarrow 2 \rightarrow 3 \rightarrow 9$, $1 \rightarrow 5 \rightarrow 9$ and $1 \rightarrow 6 \rightarrow 7 \rightarrow 8 \rightarrow 9$). In such a situation, our strategy selects the working lightpath, $1 \rightarrow 2 \rightarrow 3 \rightarrow 9$, as a pair of the target lightpath because it holds three conflicts. After the traffic on the working lightpath is switched into the target lightpath, three wavelength resources are available and used for the other target lightpaths.

4 Evaluation

Our algorithm selects a working lightpath heuristically when the lightpath is required to delete in a SWITCH or DELETE procedure. We evaluate the effectiveness of the heuristic selection at first.

4.1 Performance of the reconfiguration algorithm

We evaluate our reconfiguration algorithm with a wide area network model and various degrees of wavelengths.

4.1.1 Evaluation model

Here, we explain our evaluation model. We use the NSFNET, which has 14 nodes and 21 links, as a network model. This network topology is shown in Figure 4. We generate series of traffic matrices $T^1, T^2, \cdots, T^k$, where the elements of each traffic matrix are set a random value between zero and the transmission capacity of a fiber.

We evaluate the performance of our algorithm with logical topologies where the patterns are changed randomly since this is the most difficult case to reconfigure logical topologies without traffic loss. We examine the performance of our algorithm in the worst case. In this paper, we set $k = 5$. We simulate reconfigurations when the degree of wavelength multiplexing is 16, 32, 64, 128, or 256.

4.1.2 Logical topology design algorithm

To generate the logical topologies for those traffic matrices, we use a simple design algorithm (SDA). SDA works as follows. Given a traffic matrix, SDA selects a node pair which has a largest traffic demand in the traffic matrix. Then, SDA sets up a primary lightpath and a backup path between the nodes, and reduces the transmission capacity of a lightpath from the traffic demand (in this paper, we set the transmission capacity of a lightpath as 10 Gbps). Each of primary and backup lightpaths is selected on the shortest route, in terms of the propagation delay, among currently available routes. Backup lightpaths are selected on a disjoint sets of links of corresponding working lightpaths. And SDA also deals with shared protection strategy. For each lightpath, its wavelength is assigned based on First-Fit policy. If SDA fails to set up a primary and a backup lightpaths for the traffic demand from a source node to a destination node, SDA sets the traffic demand zero.

The properties of the logical topologies obtained by SDA are shown in Table 1. The average utilization of wavelength resources on links is 95%. We do not consider the traffic demands which are not accommodated by the logical topology design algorithm because only the loss of traffic during reconfiguration is our concern.

4.1.3 Effectiveness of heuristic selection strategy

We examine the effectiveness of those three strategies by the number of times of DELETE procedure calls (i.e., the number of traffic loss occurrences) in a reconfiguration. The results are shown in Figure 5 where the average numbers of DELETE procedure calls are dependent on the degree of wavelength multiplexing. This figure shows traffic loss of the algorithm with a heuristic strategy is more than twice as less as those of the algorithms with longest/shortest-strategy. The heuristic strategy is effective and thus we will obtain the further results by adopting the heuristic strategy.

4.1.4 Effectiveness of procedures

Next, we consider four algorithms for comparison: Algorithms 1, 2, 3, and 4. Algorithm 4 is our proposed algorithm, and the first three algorithms are subsets of Algorithm 4. They are compared in Table 2.

We also examine the performances of those four algorithms by the number of times of DELETE procedure calls in a reconfiguration (i.e., the number of traffic loss occurrences). As seen from Figure 6, traffic loss decreases in order from Algorithm 1 to 2 to 3 to 4, and there is a relatively large gap between Algorithms 2 and 3. This result shows that wavelength re-allocation is useful in reconfigurations of logical topologies. On the other hand, the BACKUP procedure is still less effective than wavelength re-allocation. We believe that the shared protection strategy makes the BACKUP procedure less effective.

4.2 Effectiveness of reconfiguring logical topologies

We had another evaluation to examine the effectiveness of reconfiguring logical topologies. In this subsection, we observed the performance of our reconfiguration algorithm when the dynamic changing of traffic model is applied and the changing of the average utilization of wavelength resources on links.

4.2.1 Traffic transition model

We have generated a series of traffic matrices $T^1, T^2, \cdots, T^k$, where those elements are random numbers between zero and the transmission capacity of a fiber. Here, we use another traffic model similar to the model described in Ref. [12]. Eq. (1) shows the traffic model. $T^1, T^2, \cdots, T^k$ are the same traffic matrices used in the above simulation. The parameter h gives h-step traffic transitions between T^i and T^{i+1}.

$$T^{k,l} = round\left[(1-\frac{l}{h})T^k + \frac{l}{h}T^{k+1}\right] \tag{1}$$

4.2.2 Consideration of the effective reconfiguration

We use NSFNET (shown in Figure 4) as a network model to show the effect of parameter h. The degree of wavelength multiplexing is 128 or 256. We set $k = 5$ and $h = 1, 4, 8$. Logical topologies are generated by applying SDA to traffic matrices $T^{m,l}$ $(1 \leq m \leq k,\ 0 \leq l \leq h)$. We examine the performances of two algorithms, Algorithms 3 and 4, by changing the average utilization of wavelength resources from 30% to 96% by a step of 2%. We realize the target utilization of wavelength resources in the logical topologies by inserting the check process of wavelength usage in SDA.

The results are shown in Figs. 7 and 8. The vertical axis shows the average number of traffic loss occurrences and the cross axis is the average of total bit rate carried in the whole network, where the bit rate is obtained by the number of primary lightpaths times the transmission capacity of the lightpath. According to Figs. 7.a. and 8.a., the number of lightpaths that Algorithm 4 can reconfigure without traffic loss is about one and a half times as much as the number of lightpaths Algorithm 3 can do. However, when h gets larger, i.e., when logical topologies change gradually, the difference of the performance between Algorithms 3 and 4 becomes smaller. In the figures, when $h = 8$, the results of Algorithms 3 and 4 are almost the same. It indicates that the BACKUP procedure is much less efficient when the utilization of wavelength resources is high and when the change of logical topologies are small.

To explain the differences which result from changing values of h, we compare the number of procedure calls in Table 3. From this table, the number of the SWITCH procedure calls also becomes larger as the value of h becomes larger, whereas the number of the APPEND procedure calls decreases. The SWITCH procedure protects traffic on working lightpaths, and thus the number of traffic loss occurrences becomes to a few or zero. We illustrate this situation in Figure 9. When h is 1, each element of the traffic matrices is changed randomly. Therefore, there may be many wavelength resource conflicts between working lightpaths and target lightpaths. Suppose that two working lightpaths P_1 and P_2 require a certain wavelength on the links 2-3 and 3-4, respectively, and a target lightpath, P_3, requires the same wavelength on the link 2-4. In this case, $APPEND(P_3)$ is tried at first. Then, $BACKUP(P_1)$ and $BACKUP(P_2)$ are tried next. If these trials do not succeed, $RELEASE(P_1)$ and $RELEASE(P_2)$ are executed. Finally, $DELETE(P_1)$ and $DELETE(P_2)$ are executed. Thus, the number of APPEND, BACKUP, RELEASE and DELETE procedure calls become larger.

On the other hand, if h is large, a gradual change on the traffic matrices results in the requesting almost the same number of lightpaths between node pairs. In Figure 9.b., the working lightpath P_2 and the target lightpath P_4 are assigned to the same route and wavelength. Therefore, P_2 remains as the target lightpath P_4, and need not be reconfigured. Or the route of the target lightpath P_3 may be selected on the different route from that used by the working lightpath P_1. In this case, $SWITCH(P_1, P_3)$ is executed. Therefore, the number of SWITCH procedure calls gets larger and other procedure calls decrease.

We go back to Figs. 7 and 8. The results show that changing to the very different logical topology, which may be an optimal logical topology, will be tolerable against the moderate utilization of wavelength resources. However, for higher utilization of wavelength resources, the gradual changes on the traffic matrices significantly reduce the traffic loss during the reconfiguration. Although Algorithm 4 reduces the number of DELETE procedure calls compared to Algorithm 3, we need a logical topology design algorithm which generate a sub-optimal topology with less changes on lightpaths, rather than the optimal one. To investigate such a design algorithm is beyond the scope of this paper and constitutes the scope of future work.

5 Summary

In this paper, we proposed an algorithm to reconfigure logical topologies in reliable WDM mesh networks. The algorithm is composed of five procedures to set up or tear down lightpaths. We first evaluated the performance of the algorithm with randomly generated traffic, and then evaluated with changes of network load using a dynamic traffic model. The results have shown that changing to the very different logical topology will be tolerable for the moderate utilization of wavelength resources. We also found that the gradual changes on the traffic demand significantly reduce the traffic loss during the reconfiguration. The objective of our current and future work is to investigate the logical topology design algorithm which generates sub-optimal topology with less changes on lightpaths.

Acknowledgements

This work was supported in part by the Telecommunications Advancement Organization of Japan (TAO), and by a Grant-in-Aid for Scientific Research (A) 14208027 from the Ministry of Education, Culture, Sports, Science and Technology of Japan.

References

[1] R. Ramaswami and K. N. Sivarajan: Design of logical topologies for wavelength-routed optical networks, IEEE Journal on Selected Areas in Communications, vol. 14, no. 5, (June 1996), pp. 840–851.

[2] R. Dutta and G. N. Rouskas: A survey of virtual topology design algorithms for wavelength routed optical networks, Optical Network Magazine, vol. 1, no. 1, (Jan. 2000), pp. 73–89.

[3] A. Gençata and B. Mukherjee: Virtual-topology adaptation for WDM mesh netoworks under dynamic traffic, Proc. of IEEE INFOCOM '02, (New York, NY, June 2002), vol. 1, pp. 48–56.

[4] D. Banerjee and B. Mukherjee: Wavelength-routed optical networks: linear formulation, resource budgeting tradeoffs, and a reconfiguration study, IEEE/ACM Transactions on Networking, vol. 8, no. 5, (Oct. 2000), pp. 598–607.

[5] M. Murata: Challenges for the next–generation Internet and the role of IP over photonic networks, IEICE Transactions on Communications, vol. E83-B, no. 10, (Oct. 2000), pp. 2153–2165.

[6] S. Ramamurthy and B. Mukherjee: Survivable WDM mesh networks, part I – protection, Proc. of IEEE INFOCOMM '99, (New York, NY, Mar. 1999), vol. 2, pp. 744–751.

[7] S. Arakawa and M. Murata: On incremental capacity dimensioning for reliable IP over WDM networks, Proc. of Opticomm ’01, (Denver, CO, Aug. 2001), vol. 4599, pp. 153–162.

[8] G. N. Rouskas and M. H. Ammar: Dynamic reconfiguration in multihop WDM networks, Journal of High Speed Networks, vol. 4, no. 3, (June 1995), pp. 221–238.

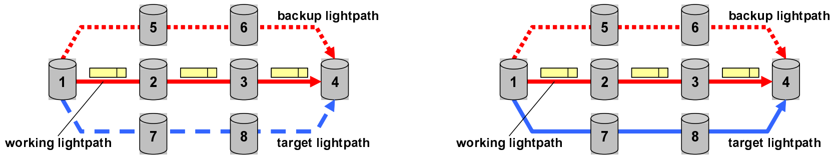
[9] J.-F. P. Labourdette and A. S. Acampora: Logically rearrangeable multihop lightwave networks, IEEE Transactions on Communications, vol. 39, no. 8, (Aug. 1991), pp. 1223–1230.

[10] I. Baldine and G. N. Rouskas: Dynamic load balancing in broadcast WDM networks with tuning latencies, Proc. of IEEE INFOCOM ’98, (San Francisco, CA, Mar.–Apr. 1998), vol. 1, pp. 78–85.

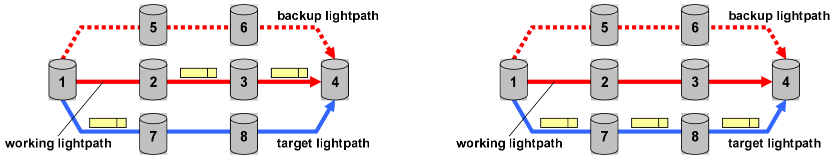
[11] J.-F. P. Labourdette, G. W. Hart, and A. S. Acampora: Branch-exchange sequences for reconfiguration of lightwave networks, IEEE Transactions on Communications, vol. 42, no. 10, (Oct. 1994), pp. 2822–2832.

[12] A. Narula-Tam and E. Modiano: Dynamic load balancing in WDM packet networks with and without wavelength constraints, IEEE Journal on Selected Areas in Communications, vol. 18, no. 10, (Oct. 2000), pp. 1972–1979.

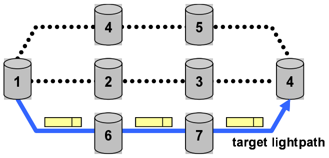
List of Figures



a. Step 1 b. Step 2

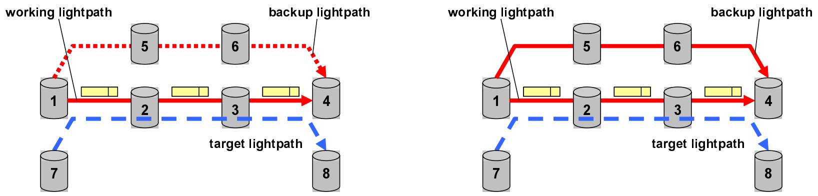


c. Step 3 d. Step 4

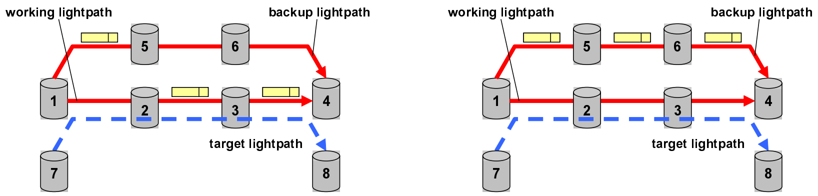


e. Step 5

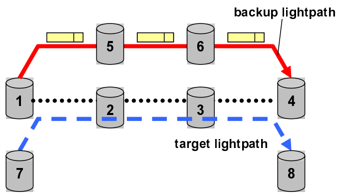
Figure 1: Each step of the SWITCH procedure



a. Step 1 b. Step 2



c. Step 3 d. Step 4



e. Step 5

Figure 2: Each step of the BACKUP procedure

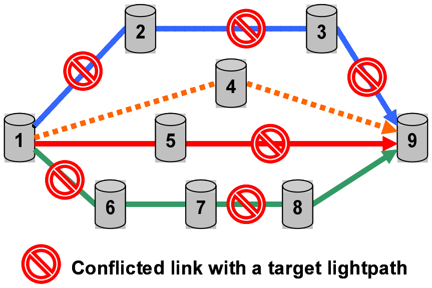


Figure 3: Heuristic selection strategy

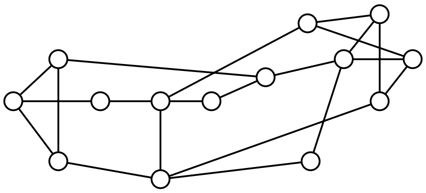


Figure 4: NSFNET (14 nodes and 21 links)

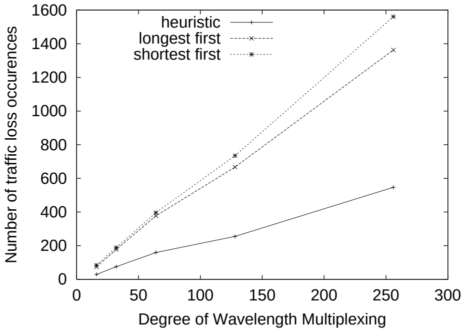


Figure 5: Number of traffic loss occurrences during a reconfiguration of a logical topology in the NSFNET.

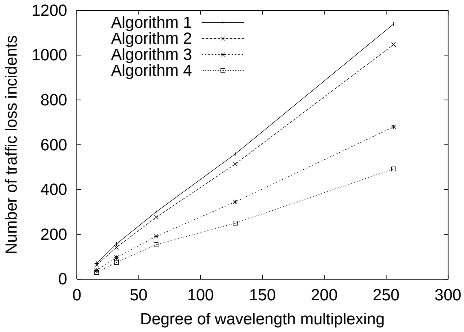
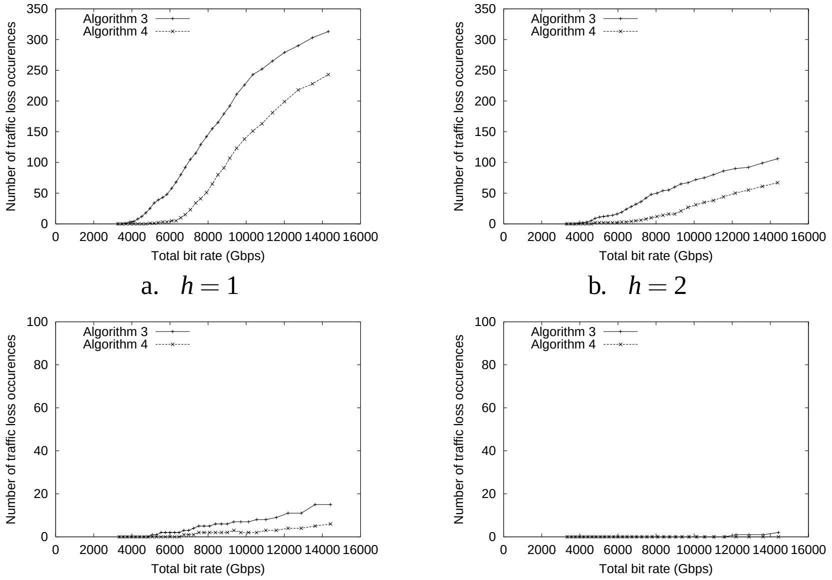


Figure 6: Number of traffic loss occurrences during a reconfiguration of a logical topology in the NSFNET.



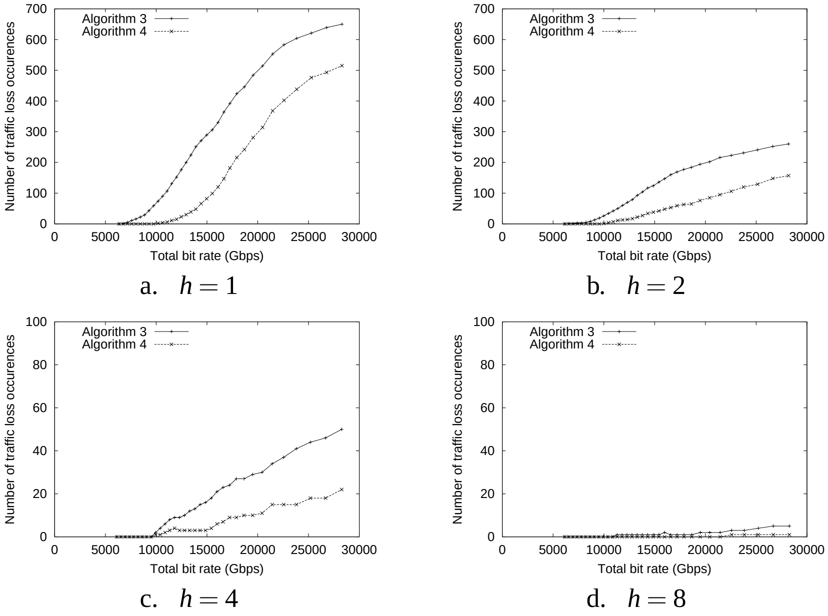
a. $h = 1$

b. $h = 2$

c. $h = 4$

d. $h = 8$

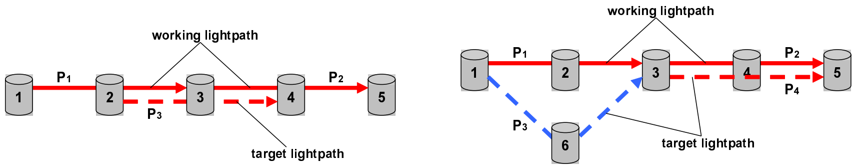
Figure 7: Relation between the performance of reconfiguration algorithms and the bit rate of the whole of logical topology when the degree of wavelength resources is 128.



a. $h = 1$ b. $h = 2$

c. $h = 4$ d. $h = 8$

Figure 8: Relation between the performance of reconfiguration algorithms and the bit rate of the whole of logical topology when the degree of wavelength resources is 256.



a. Typical case when $h = 1$ b. Typical case when $h > 1$

Figure 9: Illustrative examples of the effect of parameter h. The working lightpaths are to be disrupted and the target lightpaths are set up in a reconfiguration.

List of Tables

Table 1: Properties of logical topologies to be reconfigured. The second row shows the average number of lightpaths excluding backup lightpaths in a logical topology.

Number of wavelengths	16	32	64	128	256
Number of primary lightpaths	210	404	779	1527	3082

Table 2: Comparison of the four algorithms. Algorithm 1 is a basic algorithm composed of the SWITCH, RELEASE, APPEND, and DELETE procedures. Algorithm 2 allows the BACKUP procedure, whereas Algorithm 3 allows wavelength re-allocation. Algorithm 4 is the proposed algorithm, which employs all procedures and wavelength re-allocation.

Algorithm	*1*	*2*	*3*	*4*
SWITCH	Enabled	Enabled	Enabled	Enabled
APPEND	Enabled	Enabled	Enabled	Enabled
BACKUP	–	Enabled	–	Enabled
RELEASE	Enabled	Enabled	Enabled	Enabled
DELETE	Enabled	Enabled	Enabled	Enabled
re-allocation	–	–	Enabled	Enabled

Table 3: Number of each procedure calls in a reconfiguration with Algorithm 4 when the rate of use of wavelength resources is 96% in NSFNET

waves	*h*	*SWITCH*	*APPEND*	*BACKUP*	*RELEASE*	*DELETE*	*total*
128	1	373	913	110	973	243	2512
128	8	932	164	11	131	1	2212
256	1	727	1839	226	1961	495	4995
256	8	1856	329	22	282	1	4449

Send correspondence and proofs to Shinya Ishida.

Address: Advanced Network Architecture Laboratory
Department of Information Networking
Graduate School of Information Science and Technology, Osaka University,
Machikaneyama 1-32, Toyonaka, Osaka, 560-0043 Japan
Telephone: +81-6-6850-6863
Fax: +81-6-6850-6868
E-mail: s-isida@ist.osaka-u.ac.jp