PAPER

An integrated routing mechanism for cross-layer traffic engineering in IP over WDM networks

Yuki KOIZUMI[†], *Nonmember*, **Shin'ichi ARAKAWA**[†], *Member*, *and* **Masayuki MURATA**[†], *Fellow*

SUMMARY One approach to accommodating IP traffic on a wavelength division multiplexing (WDM) network is to construct a logical topology, establishing a set of lightpaths between nodes. The lightpaths carry IP traffic but do not require any electronic packet processing at intermediate nodes, thereby reducing the load on those nodes. When the IP and WDM networks have independent routing functions, however, the lightpaths in the WDM network may not be fully utilized by the IP router. It is therefore necessary to integrate the two routing mechanisms in order to utilize resources efficiently and adapt to changes in traffic. In this paper, we propose an integrated routing mechanism for IP over WDM networks. The key idea is to first prepare a set of *virtual-links* representing the lightpaths that can be established by the WDM network, then calculate the minimum cost route on an IP network including those links. Our simulation results show that when traffic patterns do not change, the throughput of our method is almost the same as that of a logical topology optimally designed for a given traffic demand. When traffic patterns change, the throughput of our method is about 50% higher than that of the logical topology.

key words: *wavelength division multiplexing, IP over WDM, traffic engineering, routing, integrated routing algorithm*

1. Introduction

Wavelength Division Multiplexing (WDM) technology, which carries multiple wavelength channels on a single fiber, is expected to accommodate the bulk of traffic over the current and future Internet. Since the majority of Internet traffic is IP, much research has been devoted to methods of carrying IP packets directly over a WDM network, which is an IP over WDM network [1–6]. One approach to accommodating IP traffic on a WDM network is to establish a set of lightpaths (i.e., logical links) between nodes. Each intermediate node of a lightpath has optical cross-connects (OXCs) that bind an input wavelength channel to a specified output wavelength channel. The IP packets on lightpaths are thus forwarded with no need for electronic packet processing by intermediate IP routers. In this way, lightpaths can greatly reduce the load of packet processing on the IP routers.

In order to fully utilize the bandwidth provided by WDM networks, it is important to develop new approaches to traffic engineering. Without traffic engineering, some links will be heavily congested while others are underutilized. The objective of traffic engineering can be achieved by selecting appropriate routes for balance the load on nodes and links. There are two logical topology design methods generally applied to problems of traffic engineering. One approach is the static method, which configures a logical topology based on the traffic statistics (e.g., traffic demand matrices) measured for a certain period. Given the traffic demand, an optimal logical topology can be obtained by solving a Mixed Integer Linear Problem (MILP). Some heuristic design methods can also configure nearly optimal topologies [7, 8]. Another approach is the dynamic method, which constructs logical topologies based on the current network status (e.g., link utilization).

The static method can generate the optimal logical topology for a given traffic demand matrix. It generally takes a long time to measure a traffic demand matrix, however, and the matrix becomes even more difficult to measure as the number of nodes in the network increases [9]. Moreover, due to the nature of internet traffic, even after a traffic demand matrix is obtained future traffic may not follow the same pattern. In this case, a static design method cannot flexibly adapt the logical topology to changes in the traffic pattern, which we will show in Sec. 5. One of the advantages of the dynamic method is that it does not require any information from a traffic demand matrix.

In order to configure logical topologies dynamically, a method is needed to obtain current network information such as link utilization. There are three different internetworking models for IP over WDM networks: the peer, augmented, and overlay models [10]. One of the most important differences between these models is the type of information shared between the IP layer and the optical layer. In the peer model, the network topology and all other information (e.g., routing information and link state) are shared by both layers, and a unified routing mechanism controls the whole network (i.e., both IP and WDM). In the overlay model, on the other hand, no network information is shared between the IP and WDM layers. The WDM network and IP network each have their own control planes; the routing protocols, topology information, and signaling protocols in the IP network are independent of those in the WDM network. The augmented model is a hybrid of the peer and overlay models. Some agreed-upon information such as reachability is shared between the two layers, but the two layers are managed independently as in the overlay model.

The peer model excels at efficient route control, since all network information is available. However, collecting this information requires the advertisement of the information of both the IP and WDM layers. This solution leads to an excessive update overhead, and therefore lacks scalability. In the overlay model, on the other hand, only the in-

Manuscript received January 1, 2003.
Manuscript revised January 1, 2003.

[†]The authors are with the Graduate School of Information Science and Technology, Osaka University

formation of each layer needs to be advertised. This model thus scales well compared to the peer model, but has difficulty routing packets efficiently since information on the network status is limited. Moreover, since each network has its own routing mechanism in the overlay model, lightpaths configured in the WDM network may not be fully utilized by the IP network. It is necessary to integrate the two routing mechanisms, so that resources can be used efficiently and the network can adapt to changes in traffic.

In this paper, we integrate IP routing and wavelength routing for IP over WDM networks, and propose a routing method that always forwards IP packets along lightpaths configured by the wavelength routing process. For this purpose, we introduce the concept of *virtual-links*, which are configured between IP routers. A virtual-link is a logical link that is not configured as a lightpath, but can be activated as a lightpath by requesting the required wavelength resources. In the IP network, our method first calculates routes on a topology including these virtual-links. If a virtual-link is selected as part of a route for the IP packets, a lightpath corresponding to the virtual-link will be established. In this manner, we can calculate routes on the IP network and WDM lightpaths simultaneously. The cost function for virtual-links is chosen in order to minimize the load on the nodes. The results of computer simulations show that our proposed method is effective in terms of both average end-to-end delay and throughput.

As we will describe in Sec. 2, in many previous works Multi-Protocol Label Switching (MPLS) technology has been used to specify the routes of IP packets (MPLS-based IP over WDM networks). Unlike these works, which investigate the blocking probability of Label Switched Path (LSP) requests, we intend to reveal the behavior of IP directly over WDM networks on the packet level. Since WDM technology increases the traffic that can be accommodated by a huge amount, however, computer simulations of traffic in such networks are correspondingly more difficult. We therefore also develop a simulation based on the fluid flow model, which greatly reduces the number of packets that have to be processed. We compare the results of our flow-level simulation with the results of a packet-level simulation, and show that the end-to-end packet delay is almost the same.

The rest of this paper is organized as follows. In Sec. 2, we describe how IP routing methods perform in IP over WDM networks and introduce other traffic engineering approaches. In Sec. 3, we describe our integrated routing algorithm. Before evaluating the performance of our method, we describe the flow-level simulation that enables a computer to model network traffic on a reasonable time scale. We evaluate the performance of our routing method in Sec. 5. Finally, we conclude this paper in Sec. 6.

2. Traffic engineering in IP over WDM networks

As mentioned above, it is important to develop a traffic engineering approach that can fully utilize the bandwidth provided by WDM networks. Traffic engineering objectives can be achieved by developing efficient routing methods. In this section, we describe routing methods for the overlay and peer models of IP over WDM networks.

2.1 Routing in the overlay model

In the overlay model, a logical topology is constructed as follows. First, lightpaths are established between nodes of the WDM network. Conventional IP routing protocols, such as Open Shortest Path First (OSPF) and Intermediate System to Intermediate System (IS-IS), then work on the logical topology of the WDM network. In this model, the routing protocols are not modified to account for the properties of the WDM network. The IP routing mechanism and WDM routing mechanism are designed independently, so the IP mechanism does not necessarily select the links provided by the WDM network. One typical example of such a problem is the minimum delay routing of IP networks, illustrated in Fig. 1. In this example, we assume that each optical fiber has two wavelengths, and that the propagation delay of each fiber is 1 time unit. The delay incurred by an IP router, including processing delay and queuing delay, is also 1 time unit but that of an OXC is 0. In this figure, six lightpaths are configured in the WDM network (l_1 to l_6), so the logical topology has six links. l_5, the long hop lightpath, has been configured using wavelength λ_2 between nodes N_2 and N_4. l_5 must take the longer path because wavelength λ_2 has already been used by l_2 on the optical fiber connecting nodes N_2 and N_3. There are thus two possible routes from node N_2 to N_4: R_1 and R_2. R_1 is the one-hop route $N_2 \rightarrow N_4$ using l_5, and R_2 is the two-hop route $N_2 \rightarrow N_3 \rightarrow N_4$ using l_2 and l_3. The end-to-end delay of R_1 is 4, while that of R_2 is 3. Thus, if the IP routing mechanism uses end-to-end delay as its metric, R_1 will never be used as the route from N_2 to N_4. This leads to inefficient utilization of the wavelength resources. We therefore need an integrated routing mechanism, whose main goal is to utilize resources efficiently. Some integrated routing methods have already been proposed, mainly assuming the peer model; that is, they use information from both networks such as bandwidth availability (IP) and wavelength availability (WDM). In the following section, we summarize some related works on the problem of integrated routing.

2.2 Integrated routing in the peer model

Integrated routing methods for the peer model are proposed in references [5, 6]. These papers investigate IP/MPLS over WDM networks, where the routes of IP packets can be explicitly determined by the MPLS algorithm as LSPs. In reference [5], MIRA (Minimum Interference Routing Algorithm) is proposed as an integrated routing algorithm for LSPs and lightpaths. The IP/MPLS over WDM network has a unified routing entity that collects all the topology information and link state information from both IP/MPLS and WDM networks. For incrementally arriving traffic flows where the required bandwidth is explicitly specified, MIRA

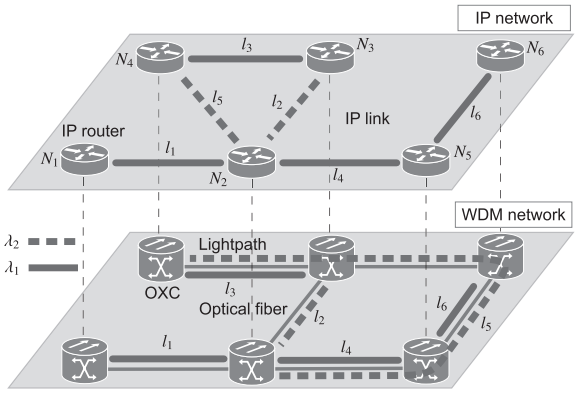


Fig. 1 An example showing that IP routing does not always select the links provided by the WDM network.

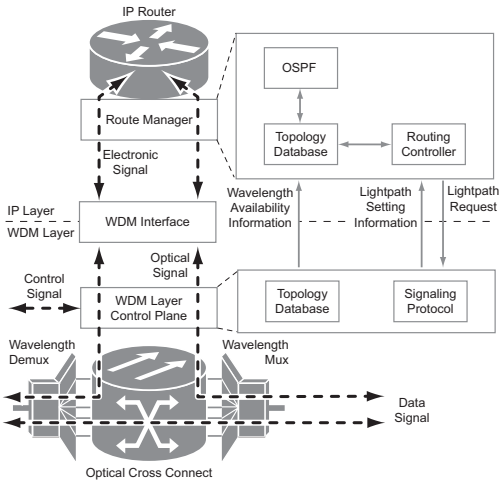


Fig. 2 Node architecture.

calculates the routes of traffic flow. MIRA also keeps track of the residual bandwidth that can be used by future LSP requests, and tries to maximize this resource. When there is no appropriate route, LSP requests are blocked. In reference [1], another integrated routing method is developed which takes into account inaccuracy in the link state information. This method incorporates the level of uncertainty in link states and hop counts into its link cost metric for LSPs. Reference [6] proposes an integrated routing method for the peer model based on the Generalized MPLS (GMPLS) framework [11]. In this work, cost values are assigned to links with the goal of reusing existing lightpaths for new LSP setting requests as much as possible.

All these works assume that IP traffic is mapped onto a series of bandwidth guaranteed LSP requests. Their main objective is to minimize the blocking probability of LSP setting requests. Few works have focused on the performance metrics of IP traffic (e.g., end-to-end delay or throughput). The works just described have all assumed a peer model network, which as mentioned above lacks scalability. Furthermore, very few works have studied integrated routing in an overlay model network. This paper therefore proposes a new integrated routing method within the overlay model of IP over WDM networks, aimed at maximizing the traffic volume that can be accommodated. In the next section, we describe this new routing method.

3. Integrated routing for cross-layer traffic engineering

In this section, we introduce our network model and the concept of a *virtual-link*, which is central to our integrated routing method. We then elaborate on our routing algorithm and discuss possible cost metrics.

3.1 Network model

We consider a network consisting of optical fiber links with W independent wavelengths and nodes that consist of IP routers with WDM interfaces, and OXCs. Figure 2 illustrates the node architecture used in this paper. The upper part of this figure shows an IP router, and the lower part shows an OXC. The IP router is connected to the OXC via a WDM interface, which converts optical signals to electronic signals and vice versa. The control plane in the WDM layer manages the topology database and wavelength availability information. The IP router has a route manager with three main blocks: the routing controller, the topology database, and OSPF. The route manager calculates routes using an integrated algorithm based on the logical topology specified in the topology database. This topology database includes only information from the IP layer such as links, router status, and network connectivity, since we assume that the internetworking architecture follows the overlay model. The OSPF-TE block advertises and collects IP link state information. Note that since we assume the overlay model, our method does not require any extension of existing IP routing protocols. The route manager in the IP layer sends setup or teardown lightpath requests to the WDM layer control plane according to the results of its route calculation. The WDM layer control plane carries out these requests and returns the results to the IP route manager.

In our network architecture, a static lightpath is set up between all adjacent nodes using one wavelength resource, to ensure end-to-end reachability. We refer to these lightpaths as *persistent lightpaths*. The other wavelength resources are used for non-persistent lightpaths, which are set up dynamically according to changes in traffic.

3.2 The concept of virtual-links

To integrate IP routing and wavelength routing, we propose a concept of *virtual-links*. A virtual-link is a logical link used by our routing algorithm. That is, they are configured on a logical topology database, and the IP routing mechanism selects routes for IP traffic from that topology. An example of a network with virtual-links is shown in Fig. 3. In this figure, persistent lightpaths are configured between all adjacent nodes, and one non-persistent lightpath is configured from router R_1 to router R_6. Three virtual-links (from router R_1 to routers R_2, R_4, and R_5) are illustrated as dashed

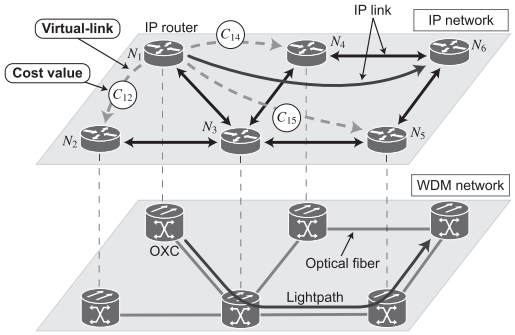
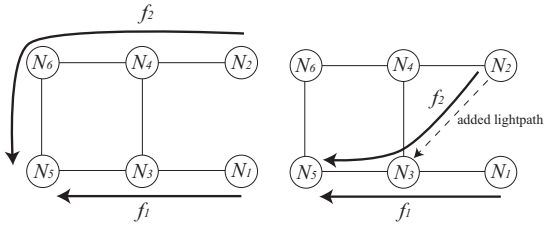


Fig. 3 A network example with virtual-links.



(a) Before configuring a lightpath (b) After configuring a lightpath

Fig. 4 Congestion resulting from activating a virtual-link.

arrows in the figure. Each link and virtual-link has an associated cost value, which is used by the IP routing mechanism. Routes are therefore calculated from a logical topology that includes virtual-links as well as existing lightpaths. Note that in our proposal, the cost of the virtual-links has a great impact on network performance. We discuss our choice of cost function for the virtual-links in Sec. 3.4.

3.3 Integrated routing algorithm

3.3.1 Topology database

The OSPF component of the IP router advertises the link state information, and each IP router collects all the link state information. The IP routers then configure following the logical topology, which includes all virtual-links and their cost.

Step 1: Set virtual-links from the source node to all destination nodes for which there is no lightpath configured.

Step 2: Assign cost values to each virtual-link using the cost function described in Sec. 3.4.

Step 3: Update the link cost values of existing non-persistent lightpaths. The same cost function described in Sec. 3.4 is used.

Step 4: Set the link cost values of persistent lightpaths to 1.

We can set the cost values of all persistent lightpaths to 1 without loss of generality, as long as the cost of virtual-links is scaled accordingly.

3.3.2 Route selection

The routes of IP packets are calculated using the topology database obtained by the above procedure, and if any virtual-links are selected as the part of IP routes then the corresponding lightpath is dynamically configured on the WDM network. Each node in the network performs following steps.

Step 1: Calculate the routes with minimum cost on the logical topology, including virtual-links and existing lightpaths, via the IP routing algorithm.

Step 2: If the resulting routes contain one or more virtual-links, send requests to the WDM network to set up the corresponding lightpaths. If some of those requests are blocked due to the lack of wavelength resources, existing lightpaths are used for those routes.

Step 3: Send teardown requests for all existing lightpaths that are not used in the IP routes just calculated. Here, an unused lightpath means that a link that is not included in the minimum cost routes just calculated, and whose source node is the node performing these steps.

Routing in the IP and WDM layers is integrated by searching for minimum cost routes in a logical topology including virtual-links. Using this method, IP traffic will be forwarded on virtual-links (i.e., lightpaths) since lightpaths are selected by the IP routing.

3.4 Cost assignment to virtual-links

The fundamental question in this approach is how to select a cost function for the virtual-links. Our main objective in choosing this cost function is to maximize the network throughput. Reducing the load on the IP routers is therefore an important criterion. Activating virtual-links as lightpaths may increase the load on some nodes, however, because more traffic will pass through the destination nodes of lightpaths. We show a simple example of this scenario in Fig. 4. In Fig. 4(a) two flows, f_1 and f_2, are forwarded along the routes $N_1 \rightarrow N_3 \rightarrow N_5$ and $N_2 \rightarrow N_4 \rightarrow N_6 \rightarrow N_5$ respectively. Figure 4(b) illustrates the network after a lightpath has been configured from node N_2 to N_3. Since the lightpath has just been set, the route of flow f_2 changes to $N_2 \rightarrow N_3 \rightarrow N_5$. In this case, f_2 traffic that used to pass through nodes N_4 and N_6 now passes through node N_3. Consequently, the load on N_3 increases. We therefore use the load on the destination node of a virtual-link as the main parameter of our cost function, to avoid concentrating traffic on frequently used nodes. Note that the IP router status can be obtained using Simple Network Management Protocol (SNMP).

The cost function C_{ij} of a virtual-link between nodes i and j is thus given by

$$C_{ij} = v_j^2 + \beta, \tag{1}$$

where v_j is the load on node j, and β is a constant offset. As noted above, our main purpose is to reduce the load on the IP routers. By using a cost function that is quadratic in v_j, we expect to strongly discourage any increase in the node load factor. By setting β to 0.5, we prevent the IP routing method from selecting routes with too many hops. Setting the offset value to 0.5 also forces the IP routing mechanism to select persistent lightpaths whenever possible, since the cost of a persistent lightpath is always 1 and the sum of the cost of two virtual-links is more than 1.

The number of lightpaths N_{ij} connecting node i to node j is decided by $N_{ij} = \lceil b_{ij} \rceil$, where b_{ij} is the incoming traffic volume at node j from node i. This assumes that the traffic volume going to node j can be measured at node i. A minimum of N_{ij} lightpaths will be configured so that the current traffic volume can be satisfied.

4. Flow-level simulation of packets in a WDM network

In this section, we describe a simulation of network traffic based on the fluid flow model (a flow-level simulation). Conventionally, traffic flow is modeled with a discrete event simulation that processes every packet (a packet-level simulation). A flow-level method is necessary, however, if simulations of large-scale networks are to finish in a reasonable amount of time. First, we detail the reasons that a flow-level simulation is needed, and then we describe the fluid flow approximation used by the simulation. Finally, we compare the results of a flow-level simulation to those of a packet-level simulation, in order to validate the method.

4.1 The difficulties of packet-level simulations

Computer simulations are commonly used to analyze and evaluate the performance of communication networks. Discrete event network simulators (such as OPNET [12] or NS-2 [13]) simulate the movement of individual packets through the network. Although this method provides accurate insight into the network state and performance, a great many events must be processed to properly simulate the network. With the growth of the Internet, the scale of networks has become even larger. Moreover, new network technologies such as WDM provide high bandwidth links, significantly increasing the traffic volume that networks can accommodate. The simulation of modern networks consumes an enormous amount of time and resources.

Many recent papers have described simulation methods based on the fluid flow model [14, 15]. These works mainly develop simulations for TCP-based networks; that is, they focus on transient states of TCP flow. In this paper, however, we only focus on the state of IP traffic over the WDM network. We therefore simplify the model of the simulation method, and just apply the fluid flow model, which requires fewer packets to be processed during simulations, and evaluate our integrated routing mechanism.

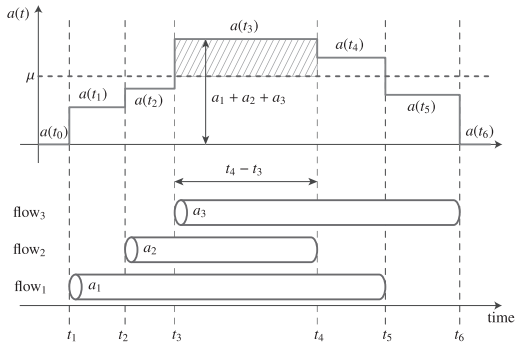


Fig. 5 Calculation of the number of packets at a node.

4.2 The flow-level simulation method

4.2.1 Fluid flow model

First, we describe the fluid flow approximation. In simulation methods based on the fluid flow model, data traffic is modeled not as a sequence of packets but as a continuous fluid. Network simulators based on the fluid flow approximation abstract the model further, focusing only on *changes* in the arrival rates of traffic flow. This can lead to a significant decrease in computation time, although any information regarding the motion of individual packets is lost.

4.2.2 Application of the fluid flow model to network simulations

We define the following terms to explain the flow-level simulation.

- t_n: n-th timestep, when the arrival or departure of flows can occur.
- $L(t_n)$: Number of packets in a node at time t_n.
- $a(t_n)$: Packet arrival rate at a node at time t_n.
- a_i: Packet arrival rate in flow i.
- μ: Service rate of the node.
- $W(t_n)$: Delay in the node at time t_n.

The packet arrival rate of each flow is drawn from a Poisson distribution with constant average rate a_i.

Figure 5 illustrates the algorithm for calculating the packet arrival rate at a node. The figure illustrates three flows (flow_1, flow_2, flow_3) that arrive at the node at consecutive timesteps t_1, t_2, and t_3. The flows, which are not all the same length, depart at times t_6, t_4, and t_5, respectively. Each flow_i has the parameter a_i, which is the arrival rate of packets in the flow_i. In other words, the time spacing between individual packets in flow_i is $1/a_i$. The packet arrival rate for a given node a (denoted $a(t_n)$) is the sum of the packet arrival rates a_i for each flow, so $a(t_n)$ changes with the arrival/departure of flows as shown in the upper part of Fig. 5.

The increase in packet number at the node from

time t_{n-1} to time t_n can be represented by $\{a(t_{n-1}) - \mu\} \times (t_n - t_{n-1})$. The factor $a(t_{n-1}) \times (t_n - t_{n-1})$ is the total number of packets arriving at the node between t_{n-1} and t_n, and the factor $\mu \times (t_n - t_{n-1})$ is the number of packets departing from the node in the same interval. Note that if $\{a(t_{n-1}) - \mu\} \times (t_n - t_{n-1}) < 0$, the number of packets in the node decreases. Since the number of packets at time t_{n-1} is denoted $L(t_{n-1})$, the total number of packets in the node at time t_n can be expressed as

$$L(t_n) = \{a(t_{n-1}) - \mu\} \times (t_n - t_{n-1}) + L(t_{n-1}).$$

Since $L(t_n)$ must be a non-negative number, we further set $L(t_n) = 0$ if the calculation results in $L(t_n) < 0$. The delay experienced at a node can be calculated as $W(t_n) = L(t_n)/a(t_n)$, applying Little's theorem [16].

In the flow-level simulation, only the head and tail packets of each flow are processed to update parameters $a(t_n)$, $L(t_n)$, and $W(t_n)$. In a packet-level simulation, every packet in the flow has to be processed; $a(t_n)$, $L(t_n)$, and $W(t_n)$ are updated whenever a packet arrives at the node.

4.3 Verification of the flow-level simulation method

To validate the fluid flow approximation, we compare the results of a flow-level simulation with those of a packet-level simulation.

We compare the two simulation methods under identical conditions: a NSFNET topology with 14 nodes and 21 edges. The number of wavelengths for each link is eight, and wavelength converters are not used in this simulation. The bandwidth of each wavelength is assumed to be 10 Gbps. The processing capacity of the IP router is set to 1 Gbps to perform the packet-level simulation within a reasonable amount of time. The flow lengths are drawn from an exponential distribution with a mean value of 12 Mb. The flows connecting each node pair ij arrive according to a Poisson process with average rate $\gamma \times d_{ij}$, where $D = \{d_{ij}\}$ is the traffic demand matrix and γ is a scale factor. We use the traffic demand matrix given in Ref. [8]. Moreover, we use the same sequence of random numbers for both simulations.

Figure 6 shows the average end-to-end delay as a function of the total amount of traffic, which is controlled by means of the scale factor γ. We observe that the average end-to-end delay obtained by the flow-level simulation agrees well with that obtained by the packet-level simulation. Although a difference appears at high loads (greater than 3.5 Gbps), the saturation point is almost the same. Networks are generally operated at loads much lower than the saturation point, so we can ignore the difference between these two methods. Evaluating the queue behavior at a node also confirms the validity of the flow-level simulation method. Due to space limitations, we omit the results of this test.

The flow-level method greatly reduces the simulation time because only two packets per flow are processed. In this validation, we chose a small average number of packets (1000) per flow, so that the packet-level simulation could run in a reasonable time. The packet-level network simulator must therefore process 500 times more packets than the flow-level simulator. The actual simulation time of the flow-level simulation for one choice of parameters was a few seconds, while that of the packet-level simulation was more than ten minutes (the exact times depend on the implementation). Moreover, the memory requirement of the flow-level simulation is much less than that of the packet-level simulation. Computer simulations based on the fluid flow model are therefore a very effective method for evaluating the performance of high-bandwidth networks.

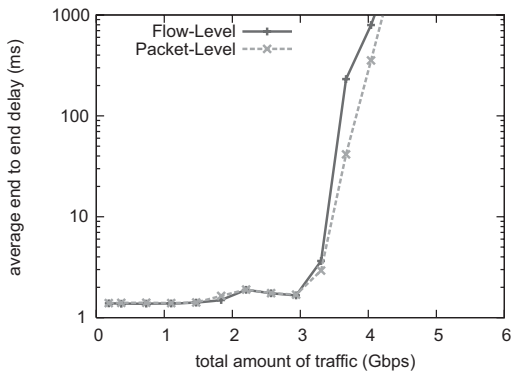


Fig. 6 The average end-to-end delay of flow-level and packet-level simulations.

5. The effects of cross-layer traffic engineering in IP over WDM networks

In this section, we evaluate the performance of the proposed algorithm through flow-level computer simulations.

5.1 Simulation model

The simulation model used here is almost the same as that described in section 4.3. We change the router capacity to 100 Gbps, the link capacity to 40 Gbps. As the number of wavelengths multiplexed on a single fiber increases, the processing capacity of electronic routers will become the bottleneck of a network. We therefore evaluate our method and conventional methods under these parameter settings, that is, the bottleneck of the network is processing capacity of electronic routers. We set the average flow length to 75 Mbytes; the packet size is 1500 bytes, and the average length of a flow is 50000 packets. In addition to the NSFNET topology, we also show results for the European Optical Network (EON), which has 19 nodes and 39 bidirectional links. For the NSFNET topology, we use a random traffic demand matrix, generated according to an exponential distribution, in addition to the matrix described in Ref. [8]. For the EON topology, we use only a randomly generated traffic matrix.

For the purpose of comparison, we use three logical topologies. First two logical topologies are designed by the algorithms SHLDA (Shortest-Hop Logical topology Design Algorithm) [7] and MLDA (Minimum delay Logical

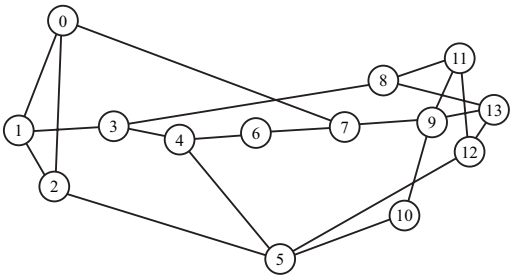


Fig. 7 NSFNET topology.

Fig. 8 European Optical Network (EON) topology.

topology Design Algorithm) [8]. These algorithms generate logical topologies based on a given traffic demand matrix in order to achieve their namesake performance objectives. Since both MLDA and SHLDA have a prior knowledge about the traffic demand matrices, they generate nearly optimal topologies. For the third logical topology, we use a full mesh topology, which is generated by configuring lightpaths between all node pairs by assuming that there are unlimited number of wavelengths on a fiber. All IP traffic in the full mesh topology is directly forwarded from source nodes to destination nodes. Since IP routers do not need to process the traffic for other node pairs, full mesh topologies are optimal and give upper bounds of networks. We compare the results of our method to these near-optimal topologies in order to demonstrate the efficiency of our method. In our simulation experiments, we use minimum hop routing for IP packets on the logical topologies generated by MLDA/SHLDA. For the full mesh topology, we configure the sufficient number of lightpaths for accommodating IP traffic between nodes.

5.2 Performance evaluations

We evaluate the performance of our integrated routing method with three metrics: the average end-to-end delay in Sec. 5.2.1, the load on the nodes in Sec. 5.2.2, and the network throughput in Sec. 5.2.3.

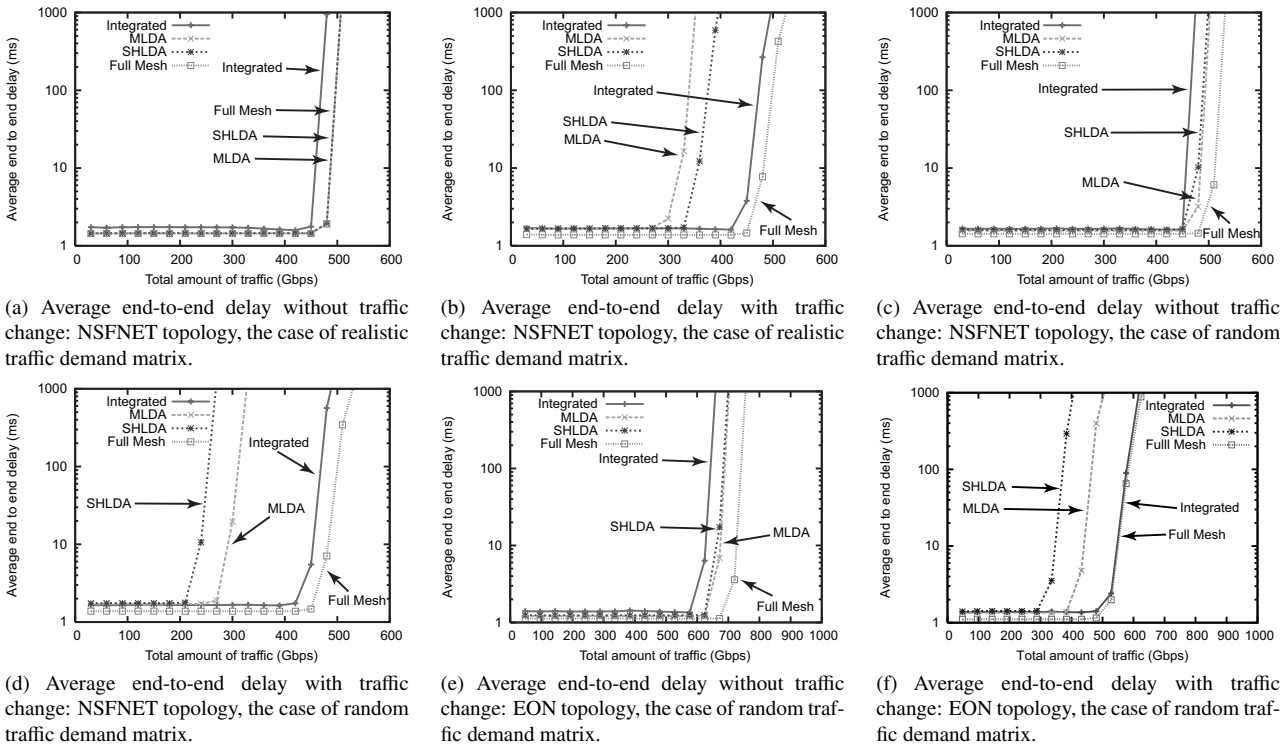
5.2.1 Average end-to-end delay

Figures 9(a) and 9(b) show the average end-to-end delay on NSFNET. The horizontal axis shows the total traffic volume in the network. The result in Fig. 9(a) is based on the traffic matrix in [8]. The average end-to-end delay achieved by our routing method is slightly worse than that achieved by the topologies generated by MLDA/SHLDA and the full mesh topology. Our method shows nearly optimal end-to-end delay performance without the information of traffic demand matrix. In Fig. 9(b), we show results from the three methods in a scenario where the traffic matrix is randomly regenerated four times in one simulation but the sum of the matrix (i.e., x-axis value in the figure) is held constant. Note that, for obtaining result of the full mesh topology, we configure the sufficient number of lightpaths between nodes for each of the regenerated traffic matrices. Thus, the results show the upper bound of the performance. As expected, changing traffic patterns degrade the performance of the SHLDA and MLDA methods drastically. Our algorithm, however, shows little increase in the average end-to-end delay in this situation. Moreover, our method does not require any prior knowledge of traffic statistics whereas SHLDA and MLDA create logical topologies according to previously measured statistics. This feature is a great advantage, since it takes a long time to measure traffic statistics accurately and IP traffic does not always follow the pattern described. Thus, statically generated logical topologies may not always perform optimally, as shown in Fig. 9(b).

We next evaluate our proposed method on the NSFNET topology using a random traffic matrix. Figures 9(c) and 9(d) show the average end-to-end delay on the NSFNET topology as a function of the total traffic. In this figure, we observe that our routing method outperforms conventional logical topology design methods. Moreover, the average end-to-end delay in our method is constant regardless of traffic change, while the delay using SHLDA or MLDA depends strongly on the traffic demand matrices. Figures 9(e) and 9(f) depict the results of our routing mechanism on the EON topology. These figures show tendencies similar to those observed on the NSFNET topology.

5.2.2 Load on nodes

To see the efficiency of our method more clearly, we show the average load on each node in Fig. 10. The load on a node is the traffic volume processed in that node divided by the processing capacity of the node. In this simulation, we use the NSFNET topology and the traffic demand matrix given in Ref. [8]. In obtaining this figure, we set the total traffic demand to the point where the average end-to-end delay just begins to increase, as seen in Fig. 9(a) and Fig. 9(b). We set the total traffic volume in the network to 450 Gbps for all methods, for both the static and dynamic traffic patterns. Figure 10(b) shows that our algorithm balances the load at an average value around 0.7 when slight congestion occurs in the network. In the SHLDA and MLDA methods, on the other hand, most of the nodes remain under-utilized even though node 12 is saturated. Fig. 10(a), however, shows that some nodes are highly loaded in all methods. To investi-



(a) Average end-to-end delay without traffic change: NSFNET topology, the case of realistic traffic demand matrix.

(b) Average end-to-end delay with traffic change: NSFNET topology, the case of realistic traffic demand matrix.

(c) Average end-to-end delay without traffic change: NSFNET topology, the case of random traffic demand matrix.

(d) Average end-to-end delay with traffic change: NSFNET topology, the case of random traffic demand matrix.

(e) Average end-to-end delay without traffic change: EON topology, the case of random traffic demand matrix.

(f) Average end-to-end delay with traffic change: EON topology, the case of random traffic demand matrix.

Fig. 9 Average end-to-end delay.

gate the reasons for this, we classify the traffic volume into three types; ingress traffic, egress traffic, and transit traffic. Ingress traffic is traffic that arrives at the node from outside the network, and egress traffic is traffic that leaves the network at the node. Transit traffic is traffic that is being forwarded from the ingress line card to the egress line card. Figure 11 shows the load on each node caused by each kind of traffic. In obtaining this result, we evaluate our method on the NSFNET topology without traffic change. It is revealed that most of the volume is either ingress or egress traffic, which must be processed at the nodes. Since the ingress and egress traffic depend on the traffic demand matrix, hereafter we focus on the behavior of the transit traffic. Note that the amount of transit traffic processed at nodes with high ingress and egress traffic (nodes 6, 7, and 12) is nearly 0. In general, the transit traffic in loaded nodes is much less than that in nodes with little load. This means that our method detours IP traffic through the less active nodes, and achieves the central goal of load balancing. SHLDA and MLDA balance the load on nodes using the information of the traffic demand matrix. Our method, on the other hand, does not require any information of traffic demand matrix, and balances the load on nodes using the information of nodal load. In the full mesh logical topology, IP routers does not need to process transit traffic, and therefore, the load on nodes is the lowest among all methods. Our method, however, shows almost the same performance as the full mesh logical topology by balancing the load.

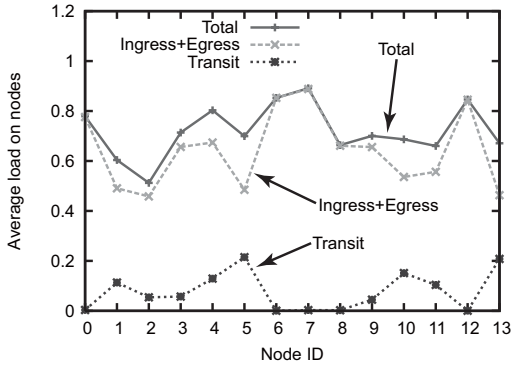
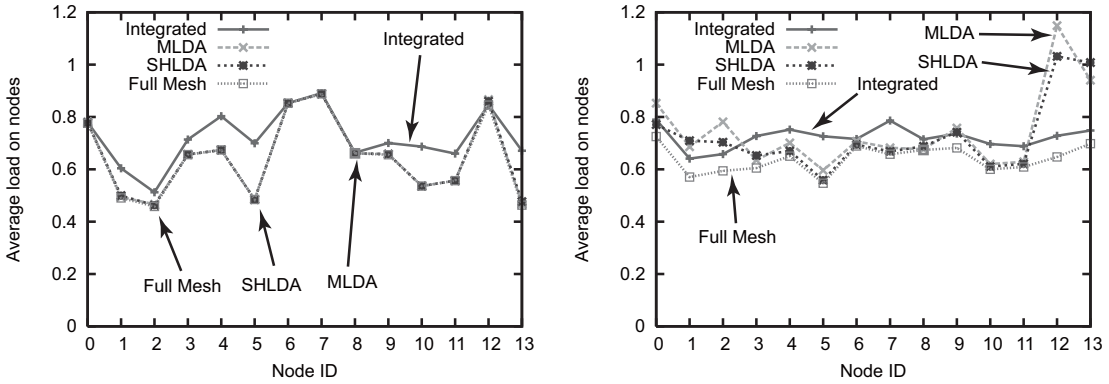


Fig. 11 Classified load on nodes without traffic change: the integrated routing method.

5.2.3 Network throughput

We next measure the network throughput achieved by cross-layer traffic engineering. We define network throughput as the maximum traffic volume that the network can accommodate while keeping the average end-to-end delay under 10 ms. This simulation also uses the NSFNET topology. Fig. 12 shows the throughput using the traffic demand matrix given in Ref. [8]. The throughput of our method is slightly lower than that of MLDA, SHLDA, and full mesh topology when traffic does not change. Changing traffic patterns, however, degrade the throughput of MLDA/SHLDA dramatically. Under these conditions, the logical topology generated by SHLDA can accommodate only 68% of the



(a) Load on nodes without traffic change: NSFNET topology, total traffic volume is 450 Gbps for all method.

(b) Load on nodes with traffic change: NSFNET topology, total traffic volume is 450 Gbps for all method.

Fig. 10 Average load on each node.

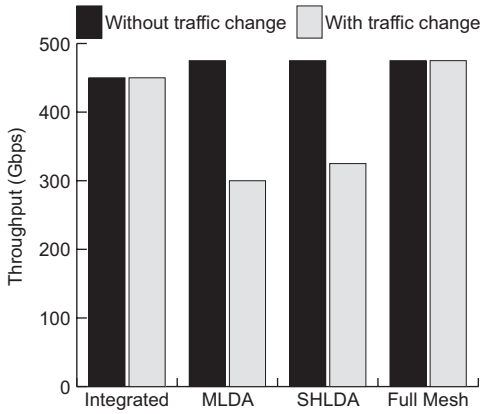


Fig. 12 Network throughput.

static traffic throughput. The logical topology generated by MLDA can accommodate 63% of the static traffic throughput. Changing traffic patterns does not greatly affect the throughput of our method.

6. Conclusion

In this paper, we propose a new integrated routing algorithm that uses virtual-links in selecting routes for IP packets, and demonstrate the necessity of dynamic lightpath configuration in IP over WDM networks. The main objectives of our algorithm are to reduce the load on IP routers and maximize the traffic volume that can be accommodated by the network. Simulation results show that our method balances the load on IP routers where static logical topology design methods cannot. Under changing traffic patterns, the throughput of our method was 38–50% higher than that of two static logical topology design methods. To reduce the costs of computer simulation, we developed a method using the fluid flow model. The flow-level simulations were validated by comparing them to conventional packet-level simulations. The flow-level and packet-level simulations achieved similar results with respect to average end-to-end delay and number of packets per node. The flow-level simulation ran about 500 times faster than the packet-level simulation and reduced the required memory, so it is a good technique for modeling large scale and high-bandwidth networks.

References

[1] 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.

[2] T. Ye, Q. Zeng, Y. Su, L. Leng, W. Wei, Z. Zhang, W. Guo, and Y. Jin, "On-line integrated routing in dynamic multifiber IP/WDM networks," *IEEE Journal on Selected Areas in Communications*, vol. 22, pp. 1681–1691, Nov. 2004.

[3] S. Arakawa, M. Murata, and H. Miyahara, "Functional partitioning for multi-layer survivability in IP over WDM networks," *IEICE Transactions on Communications*, vol. E83-B, pp. 2224–2233, Oct. 2000.

[4] N. Ghani, S. Dixit, and T.-S. Wang, "On IP-over-WDM integration," *IEEE Communications Magazine*, vol. 38, pp. 72–84, Mar. 2000.

[5] M. Kodialam and T. V. Lakshman, "Integrated dynamic IP and wavelength routing in IP over WDM networks," in *Proceedings of IEEE INFOCOM*, pp. 358–366, Apr. 2001.

[6] J. Comellas, R. Martinez, J. Prat, V. Sales, and G. Junyent, "Integrated IP/WDM routing in GMPLS-based optical networks," *IEEE Network Magazine*, vol. 17, pp. 22–27, Mar./Apr. 2003.

[7] J. Katou, S. Arakawa, and M. Murata, "A design method for logical topologies with stable packet routing in IP over WDM networks," *IEICE Transactions on Communications*, vol. E86-B, pp. 2350–2357, Aug. 2003.

[8] R. Ramaswami and K. N. Sivarajan, "Design of logical topologies for wavelength-routed optical networks," *IEEE Journal on Selected Areas in Communications*, vol. 14, pp. 840–851, June 1996.

[9] A. Medina, N. Taft, K. Salamatian, S. Bhattacharyya, and C. Diot, "Traffic matrix estimation: Existing techniques and new directions," in *Proceedings of ACM SIGCOMM*, pp. 161–174, Aug. 2002.

[10] S. Koo, G. Sahin, and S. Subramaniam, "Dynamic LSP provisioning in overlay, augmented, and peer architectures for IP/MPLS over WDM networks," in *Proceedings of IEEE INFOCOM*, pp. 514–523, Mar. 2004.

[11] E. Mannie, "Generalized multi-protocol label switching (GMPLS) architecture." RFC 3945 (Proposed Standard), Oct. 2004.

[12] "OPNET." `http://www.opnet.com/`.

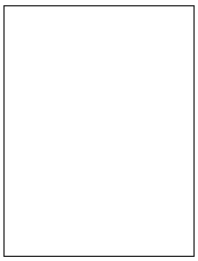
[13] "The network simulator - ns-2." `http://www.isi.edu/nsnam/ns/`.

[14] C. Kiddle, R. Simmonds, C. L. Williamson, and B. Unger, "Hy-

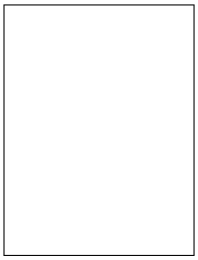
brid packet/fluid flow network simulation," in *Proceedings of the 17th Workshop on Parallel and Distributed Simulation*, pp. 143–152, June 2003.

[15] F. Baccelli and D. Hong, "Flow level simulation of large IP networks," in *Proceedings of IEEE INFOCOM*, pp. 1911–1921, Mar. 2003.

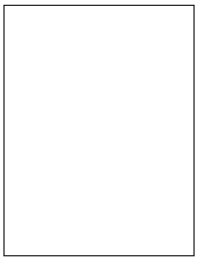
[16] D. Bertsekas and R. Gallager, *Data Networks*. Englewood Cliffs, New Jersey: Prentice Hall, Inc., Englewood Cliffs, 1987.



Yuki Koizumi received the M.E. degrees from Osaka University, Japan, in 2006. He is currently a doctoral student at the Graduate School of Information Science and Technology, Osaka University, Japan. His research interest includes traffic engineering and routing in photonic networks.



Shin'ichi Arakawa received the M.E. and D.E. degrees in Informatics and Mathematical Science from Osaka University, Japan, in 2000 and 2003, respectively. He is currently a Research Assistant at the Graduate School of Information Science and Technology, Osaka University, Japan. His research work is in the area of photonic networks. He is a member of IEEE and IEICE.



Masayuki Murata received the M.E. and D.E. degrees in Information and Computer Sciences from Osaka University, Japan, in 1984 and 1988, respectively. In April 1984, he joined IBM Japan's Tokyo Research Laboratory, as a Researcher. From September 1987 to January 1989, he was an Assistant Professor with the Computation Center, Osaka University. In February 1989, he moved to the Department of Information and Computer Sciences, Faculty of Engineering Science, Osaka University. In April 1999, he became a Professor of Osaka University, and since April 2004, he has been with the Graduate School of Information Science and Technology, Osaka University. He has contributed more than four hundred and fifty papers to international and domestic journals and conferences. His research interests include computer communication networks and performance modeling and evaluation. He is an IEICE Fellow. He is a member of IEEE, ACM, The Internet Society, IEICE and IPSJ.