

A Wavelength Rerouting Algorithm in Wide-Area All-Optical Networks

Kuo-Chun Lee, *Member, IEEE*, and Victor O. K. Li, *Fellow, IEEE*

Abstract—This paper considers rerouting and minimization of incurred disruption due to rerouting in a wide-area all-optical wavelength division multiplexed (WDM) network with random circuit arrivals and departures. One limitation of such a network is the wavelength continuity constraint imposed by the all-optical cross-connect switches which do not allow a circuit to be placed on a nonwavelength-continuous route. Wavelength rerouting is proposed to rearrange certain existing circuits to create a wavelength-continuous route in order to accommodate a new circuit. To reduce the disruption period, move-to-vacant wavelength-retuning (MTV_WR) is used as the basic operation of circuit migration, in which a circuit is moved to a vacant wavelength on the same path, and parallel MTV_WR rerouting is used to reroute multiple circuits. An optimal algorithm is developed to minimize the weighted number of rerouted circuits with parallel MTV_WR rerouting. In our test network, wavelength rerouting can effectively alleviate the wavelength continuity constraint by reducing call blocking probability an average of 30% while reducing the number of rerouted circuits and the disruption period.

I. INTRODUCTION

ALL-OPTICAL networks transmit information on a very high-speed, broadband optical path [1]–[3]. One promising all-optical wide-area network architecture is the wavelength division multiplexed (WDM) network with circuit-switching and wavelength routing wherein a few wavelength cross-connect switches are used to connect a sequence of wavelength channels between the information source and the destination to set up a circuit for data transmission [4]. WDM allows parallel transmission with low-speed devices, circuit switching eliminates the need of optical buffering, and wavelength routing reuses wavelengths.

One possible implementation of the all-optical cross-connect switches is to use wavelength demultiplexers to separate the inbound wavelengths and then use directional couplers to switch various wavelengths to the desired output links. In this switching process, a message must maintain its transmission wavelength as it is forwarded in the network. This is called the *wavelength continuity constraint* [5], [6]. Wavelength continuity reduces channel utilization because a nonwavelength-

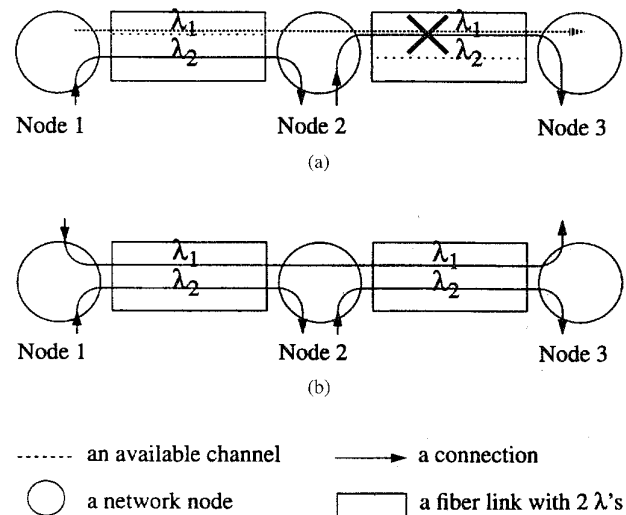


Fig. 1. An example where rerouting improves wavelength utilization.

continuous route cannot be used to set up a circuit for data transmission although the route is available. This is especially severe in a circuit-switched network with random circuit arrivals and departures because the unpredictable traffic demand renders the use of routing to mitigate the wavelength continuity constraint ineffective.

To alleviate the inefficiency due to the wavelength continuity constraint in a circuit-switched network with random circuit arrivals and departures, wavelength rerouting is used. Wavelength rerouting changes the wavelengths of existing circuits and makes the available route wavelength continuous, thereby increasing channel utilization. For example, in Fig. 1, a new circuit can be set up on a wavelength-continuous route from node 1 to node 3 after the existing circuit $u_{2 \rightarrow 3}$ (the circuit from nodes 2–3) on wavelength λ_1 is rerouted to λ_2 . The advantage of rerouting is increased carried traffic for a given circuit blocking probability or, equivalently, reduced circuit blocking probability for a given offered traffic.

While rerouting can be used to improve network utilization, there will be low or even zero throughput during the rerouting process. Transmissions of the rerouted circuits must be temporarily shut down to prevent data from being lost or misrouted. The throughput loss is particularly pronounced in all-optical networks wherein each circuit on a dedicated, broadband wavelength channel is expected to carry gigabits

Manuscript received April 17, 1995; revised October 20, 1995. This work was supported by the NSF under Grant NCR-9318497.

K.-C. Lee is with Qualcomm Incorporated, San Diego, CA 92121 USA.

V. O. K. Li is with Department of Electrical Engineering, Communication Sciences Institute, University of Southern California, Los Angeles, CA 90089 USA.

Publisher Item Identifier S 0733-8724(96)04081-9.

of information flow, and hence a tiny period of outage on a single circuit will cause significant data loss. Furthermore, a longer period of disruption is incurred because of the longer propagation delay for transmitting signaling messages in the wide-area network. Therefore, minimization of the incurred disruptions (e.g., the number of rerouted circuits or the disruption period of existing connections) is imperative for rerouting in all-optical, wide-area networks.

Some research has been published on minimization of disruptions during reconfiguration. Reference [7] considered the reconfiguration in topology patterns of a broadcast-and-select lightwave network [4]. The reconfiguration is based on a sequence of branch-and-exchange (BE) operations where each BE only allows two pairs of tunable transmitters and receivers to swap their wavelengths to avoid disruptions on traffic. An algorithm is developed to minimize the number of BE operations for the network to reconfigure from the current to the target topology.

To minimize the incurred disruption in our wavelength rerouting, we design a new circuit migration scheme for rerouting circuits to greatly reduce the disruption period and an algorithm to select the fewest number of circuits to be rerouted. Move-to-vacant wavelength-retuning (MTV_WR) is used as the basic operation of circuit migration, in which a circuit is moved to a vacant wavelength on the same path (a path is a sequence of loop-free communication links), and the parallel MTV_WR rerouting scheme is used to reroute multiple circuits on disjoint sets of links. An optimal rerouting algorithm is sought to select a minimum weighted number of circuits to be rerouted with the parallel MTV_WR rerouting scheme.

The rest of this paper is organized as follows. Section II describes the system configuration and the routing algorithm. Section III describes the basic operations of circuit migration and the rerouting scheme. Section IV presents the rerouting algorithm for rerouting the existing circuits and routing a new connection. Section V shows the performance results. Section VI concludes this paper.

II. SYSTEM CONFIGURATION

An optical network with topology $G(N, L)$ is considered where N is the set of communication nodes and L the set of communication links. The bandwidth of each optical fiber communication link is wavelength-division multiplexed into a set of w wavelengths, $\Lambda = \{\lambda_1, \lambda_2, \dots, \lambda_w\}$. Each node is assumed to have a sufficiently large number of optical transmitters and receivers to service the connections such that a new connection request will not be blocked due to a lack of optical transmitters or receivers (in the worst case, this assumption may require as many transmitters/receivers as outbound/inbound channels at a particular node).

The new connection requests will arrive randomly as a Poisson process and, once accepted, will hold the circuits for exponentially distributed times with means much larger than the network-wide propagation delay and the connection setup delay. Each circuit will occupy a single dedicated wavelength in the discrete wavelength set Λ when it passes through the network; wavelength sharing by more than one connection

is prohibited. Each switching node is capable of connecting individual inbound wavelengths independently to their desired outbound links by the wavelength multiplexer/demultiplexer and the optical switch. A circuit must be routed through a set of links with the same wavelength from the origin to the destination, and a switching node cannot change the wavelength of a circuit. ([8] describes the architecture and control in an all-optical network with wavelength shifters.) A central controller with current network status information is responsible for routing the new connections in the following two phases:

- 1) route a new circuit without rerouting any existing circuits and
- 2) route a new circuit after rerouting some existing circuits when phase I fails.

As the controller receives a new connection request, phase I will be initiated to select a proper path, consisting of a sequence of links, and a proper wavelength to set up a circuit. If the initial attempt fails, the routing controller will execute phase II. It will determine which existing circuits should be rerouted and how they should be rerouted in order to accommodate the new connection and select the best path and wavelength to set up a new circuit if rerouting is feasible. Then the controller will send control information to certain switching nodes for setting up a new circuit and rerouting the existing circuits. Finally, it will set up the new connection. If rerouting is infeasible, the new connection request is rejected. (A rejected connection is assumed to be cleared in the simulation.)

In the following, the algorithm for phase-I routing is described.

A. Routing Algorithm

Solving the routing problem optimally has been shown to be computationally intractable [5] because a huge number of feasible solutions exist. We modify the algorithm in [5] as the routing algorithm. The approach is to transform the network to a graph. The weight label on each edge (link) represents the current link status and the cost of routing a new circuit on that link. To reduce the computational complexity and to simplify the notation, we decompose the graph into a few disjoint subgraphs, each corresponding to the network on a particular wavelength. Breaking the graph into subgraphs can possibly allow a parallel computation of the routing algorithm. The routing algorithm finds the shortest path on each subgraph and then chooses the least costly one among all the individual subgraphs. Smallest wavelength index is used to break a tie. The minimum-cost wavelength and its associated shortest path will be the solution. A few notations are defined. Let P_{ik} , $i \neq k$, be the set of all paths connecting nodes i to k on the network $G(N, L)$ and $P = \cup_{i, k \in N, i \neq k} P_{ik}$ be the set of all possible paths. A route p^λ is a path $p \in P$ using a wavelength $\lambda \in \Lambda$. U is the set of existing circuits in the network at the time of interest, and u_0 is the new connection request (the time index is dropped for simplicity). The algorithm transforms the network to a graph, labels the edges of the graph with weights, and searches for the best route.

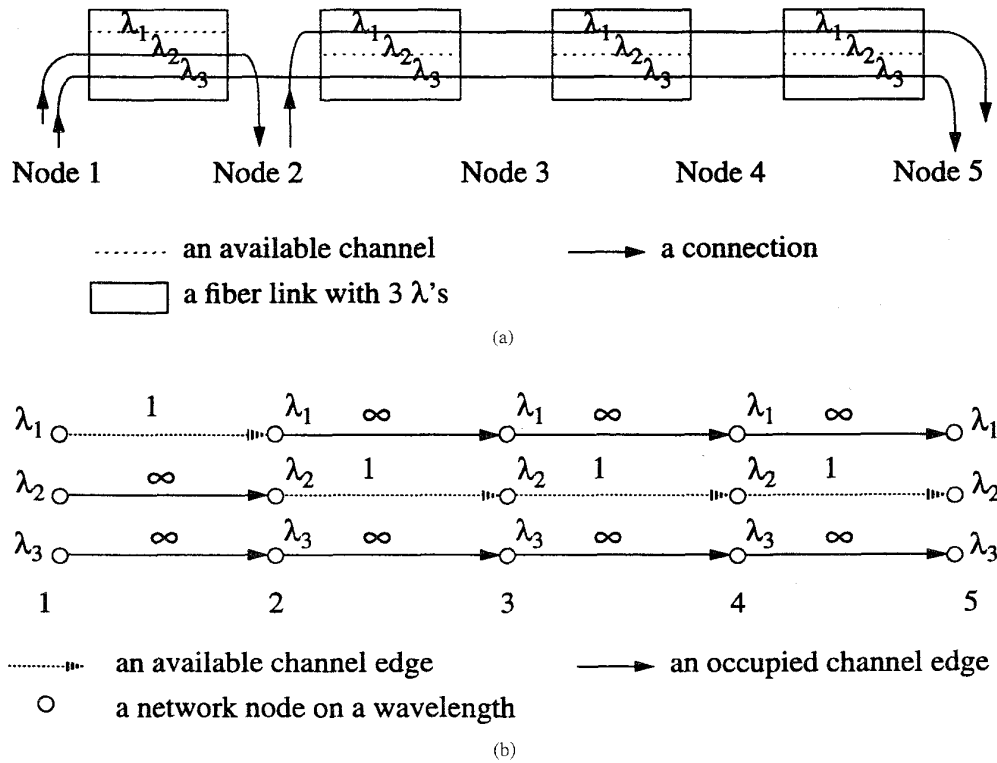


Fig. 2. Illustration of the graph transformation of phase I of the routing algorithm.

1) *Graph Transformation*: The network $G(N, L)$ with the wavelength set Λ is transformed into a collection of disjoint subgraphs $G(N^\lambda, L^\lambda)$, $\lambda \in \Lambda$, each corresponding to the network on a particular wavelength. For each wavelength $\lambda \in \Lambda$, the subgraph $G(N^\lambda, L^\lambda)$ is obtained by generating a vertex i^λ and an edge (i^λ, j^λ) if $i \in N$ and $(i, j) \in L$, respectively. Thus the new graph is $\cup_{\lambda \in \Lambda} G(N^\lambda, L^\lambda)$ where

$$N^\lambda = \{i^\lambda : i \in N\}, \lambda \in \Lambda,$$

$$L^\lambda = \{(i^\lambda, j^\lambda) : (i, j) \in L\}, \lambda \in \Lambda.$$

Fig. 2 shows a typical example. Decomposing the given network into wavelength-disjoint subgraphs is dictated by the wavelength continuity constraint which restricts a feasible route to traverse a particular subgraph rather than multiple subgraphs.

2) *Cost Labeling*: The network occupancy function $q(i^\lambda, j^\lambda) = u \in U$ if the connection u occupies wavelength λ of link (i, j) and $q(i^\lambda, j^\lambda) = NULL$ if wavelength λ of link (i, j) is idle. The cost function of each edge of the graph is determined by whether a channel is idle or busy, i.e.,

$$c(i^\lambda, j^\lambda) = \begin{cases} 1, & \text{if } q(i^\lambda, j^\lambda) = NULL \\ \infty, & \text{if } q(i^\lambda, j^\lambda) \neq NULL. \end{cases}$$

3) *Route Searching*: The best route is selected as follows.

- For each $\lambda \in \Lambda$, find the shortest path p_o^λ with cost $c(p_o^\lambda)$ on subgraph $G(N^\lambda, L^\lambda)$.

- Select the least costly route $p_o^{\lambda^o}$ as $c(p_o^{\lambda^o}) \leq c(p_o^\lambda), \forall \lambda \in \Lambda$ (smallest index of wavelength is used to break a tie).

If $c(p_o^{\lambda^o}) < \infty$, phase I is complete and $p_o^{\lambda^o}$ is the solution; otherwise phase II is needed and rerouting will be considered. In phase II, the new connection request may be accommodated by rerouting certain existing circuits while reducing disruptions. We use a circuit migration scheme to move a circuit with very short disruption period and a rerouting algorithm to minimize the weighted number of circuits rerouted. Move-to-vacant wavelength-retuning (MTV_WR) will be used for circuit migration, and parallel MTV_WR rerouting reroutes multiple circuits on disjoint sets of links. The rerouting algorithm reroutes a minimum weighted number of existing circuits and routes the new connection request.

III. CIRCUIT MIGRATION

A. Basic Operation

The basic operation of rerouting a circuit consists of wavelength-retuning (WR) and move-to-vacant (MTV).

1) *Wavelength-Retuning*: Wavelength-retuning retunes the wavelength of a circuit but maintains its path. (In general, rerouting may change the path and/or the wavelength of an existing circuit.) Wavelength-retuning has certain advantages. First, it facilitates control because the old and the new routes share the same switching nodes. Second, it reduces the calculation of optimal rerouting because only changing the wavelength of an existing circuit need be considered. Third,

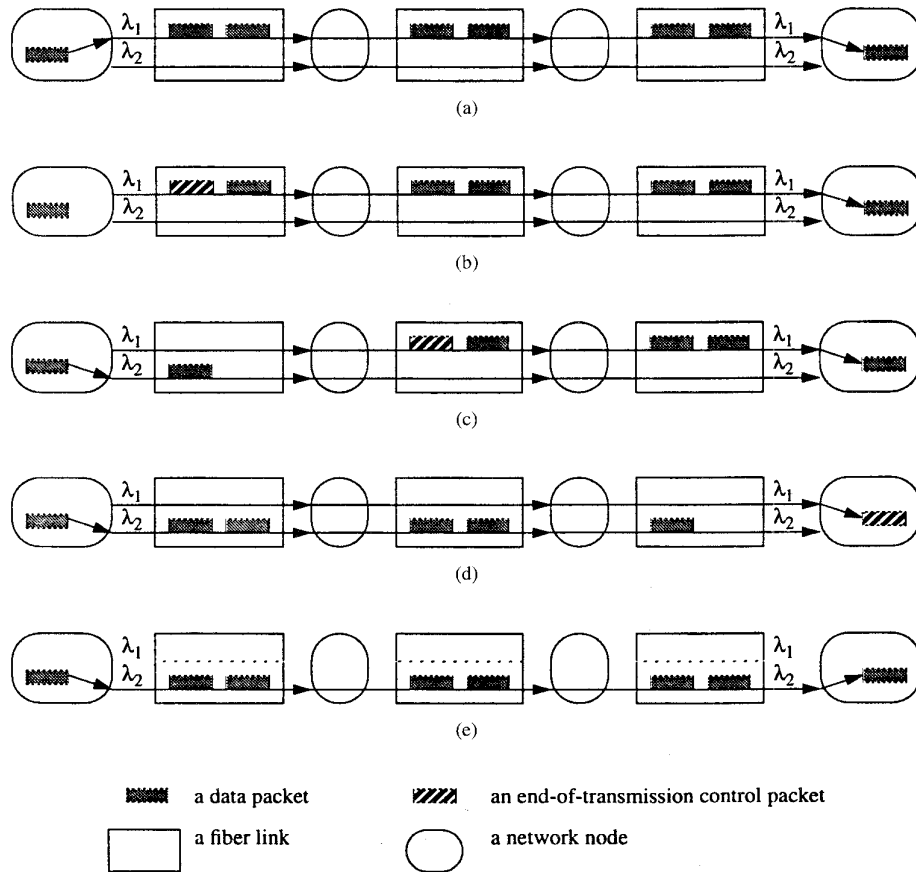


Fig. 3. Implementation of the MTV_WR operation of circuit migration.

it significantly reduces the disruption period in the rerouting phase when it is combined with move-to-vacant in a wide-area network.

2) *Move-To-Vacant*: Move-to-vacant reroutes a circuit to a vacant route with no other circuits. It has the following advantages. First, move-to-vacant does not interrupt other circuits since there is no other circuits on the new route of the rerouted circuit. Second, it preserves transmission on the old route during the setup of the new route, and therefore the disruption period is reduced.

3) *Move-To-Vacant Wavelength-Retuning*: Move-to-vacant wavelength-retuning moves a circuit to a vacant wavelength on the same path. It can greatly reduce the disruption period.

Fig. 3 illustrates how it is implemented. In (a), the network controller sends control messages to the intermediate switches on the path of the rerouted circuit to set up a new route on a vacant wavelength, e.g., by setting the state of a switch such that the new wavelength is switched from an inbound link to an appropriate outbound link. Then, the origin node prepares to switch data transmission from the old to the new wavelengths. In (b), the origin node appends an *end-of-transmission* control packet after the last packet on the old wavelength and holds the first packet on the new wavelength for a guard time (this guard time will determine the length of the disruption period). The end-of-transmission packet is used to inform the destination

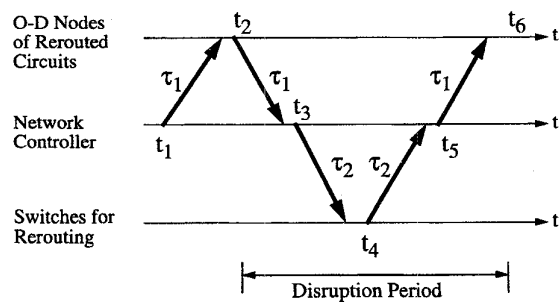


Fig. 4. Illustration of long disruption period in reroute-after-shutdown rerouting schemes.

node that data transmission via the old wavelength has ended and data will soon arrive via the new wavelength. The guard time prevents data from being lost during the transient period of circuit migration.

In (c), (d), and (e), data transmission is switched from the old to the new vacant wavelength. The information source switches transmission by retuning the wavelength of the optical transmitter to the new wavelength. At the destination node, the information sink switches reception by retuning the wavelength of the receiver to the new wavelength upon detecting the end-of-transmission control packet.

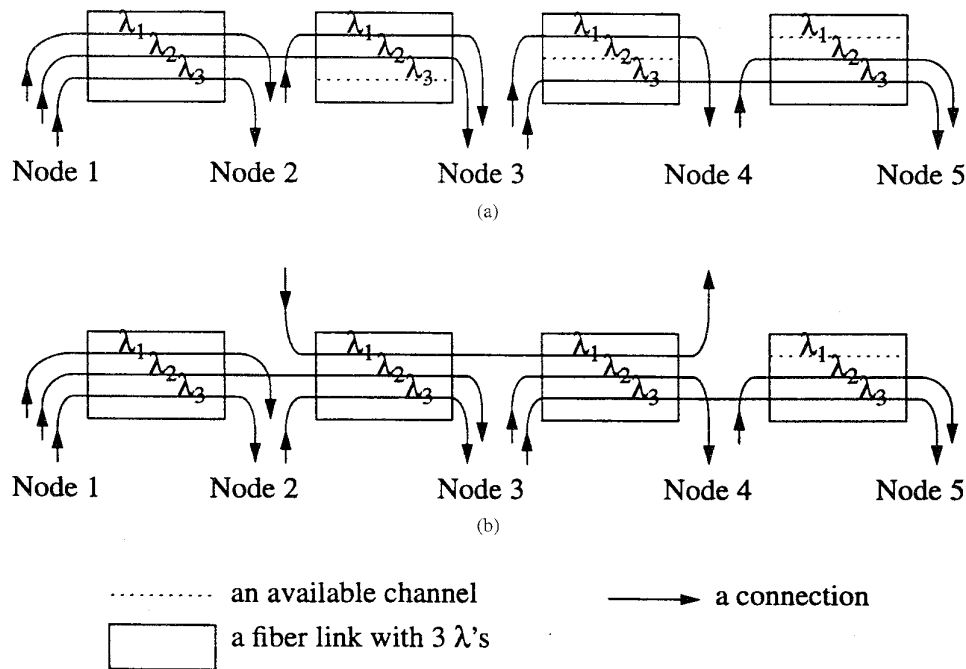


Fig. 5. Illustration of parallel move-to-vacant wavelength-retuning rerouting.

The guard time should be bounded by the sum of i) the switching times of tunable optical transmitter and receiver τ_s , ii) the processing time of detecting the end-of-transmission control packet at the destination τ_p , and iii) the differential propagation delay of sending messages through two wavelengths due to the wavelength dispersion of optical fibers or optical devices $\tau_d = \eta \Delta_\lambda R$ where η is the value of dispersion in ps/(nm \times km), Δ_λ the difference between two wavelengths in nm, and R is the distance traveled in km. For example, in a wide-area network with a diameter of 1000 km and a bandwidth of 35 nm, τ_d can be as big as 0.56 μ s ($\eta = 16$). If τ_s and τ_p are assumed to be 1 μ s, then the total guard time is only of the order of μ s. This small guard time should be the only disruption period of MTV_WR operation and the associated rerouting scheme in the next subsection.

On the other hand, *rerouting after shutdown* (RAS) incurs a long disruption period in a wide-area network. Fig. 4 shows a typical example. The network controller sends the control message to the origins and destinations of the rerouted circuits to stop transmissions at t_1 . These origins stop transmissions and send acknowledgments back to the controller at t_2 ($\geq t_1 + \tau_1$ where τ_1 is the maximum propagation delay between the network controller and the origins or the destinations of the rerouted circuits). The network controller receives all the acknowledgments from these origins and sends control messages to the switches involved in rerouting to set up the new routes at t_3 ($\geq t_2 + \tau_1$). The switches send acknowledgments of completing switching at t_4 ($\geq t_3 + \tau_2$ where τ_2 is the maximum propagation delay between the network controller and the switches involved in rerouting). The network controller receives all the acknowledgments from the switches and sends

control messages to the origins and destinations for restarting transmissions at t_5 ($\geq t_4 + \tau_2$). Finally the origins resume the transmissions via the new route at t_6 ($\geq t_5 + \tau_1$). Thus, the transmissions of the rerouted circuits will be disrupted for $t_6 - t_2 \geq 2\tau_1 + 2\tau_2$. Therefore, the period of disruption is approximately four times the single-hop propagation delay of the network.

Assuming a single-hop distance of tens of miles, the disruption period of RAS is of the order of a hundred μ s. Therefore, MTV_WR can reduce the disruption period by two orders of magnitude compared to RAS.

B. Rerouting Scheme

Using the basic MTV_WR operation for rerouting a single circuit, we can reroute multiple circuits by using the parallel MTV_WR rerouting scheme.

Parallel MTV_WR moves each of the rerouted circuits to a vacant wavelength on the same path in parallel. The rerouted circuits should be on disjoint sets of links. For example, in Fig. 5, the existing circuits $u_{2 \rightarrow 3}$ and $u_{3 \rightarrow 4}$ pass through the disjoint links (2, 3) and (3, 4), respectively, and are retuned simultaneously to the vacant wavelengths, λ_3 and λ_2 . Then the wavelength λ_1 is released to accommodate the new connection request $u_{2 \rightarrow 4}$. The overall delay of executing the parallel MTV_WR rerouting scheme is the maximum of executing each of the MTV_WR operations required, and therefore it should be quite small.

Parallel MTV_WR rerouting, however, cannot always be useful. For example, in Fig. 6, the existing circuit $u_{1 \rightarrow 2}$ is first rerouted to λ_3 and releases its wavelength λ_2 . Then the existing circuit $u_{1 \rightarrow 3}$ is rerouted to a vacant wavelength λ_2 . Finally,

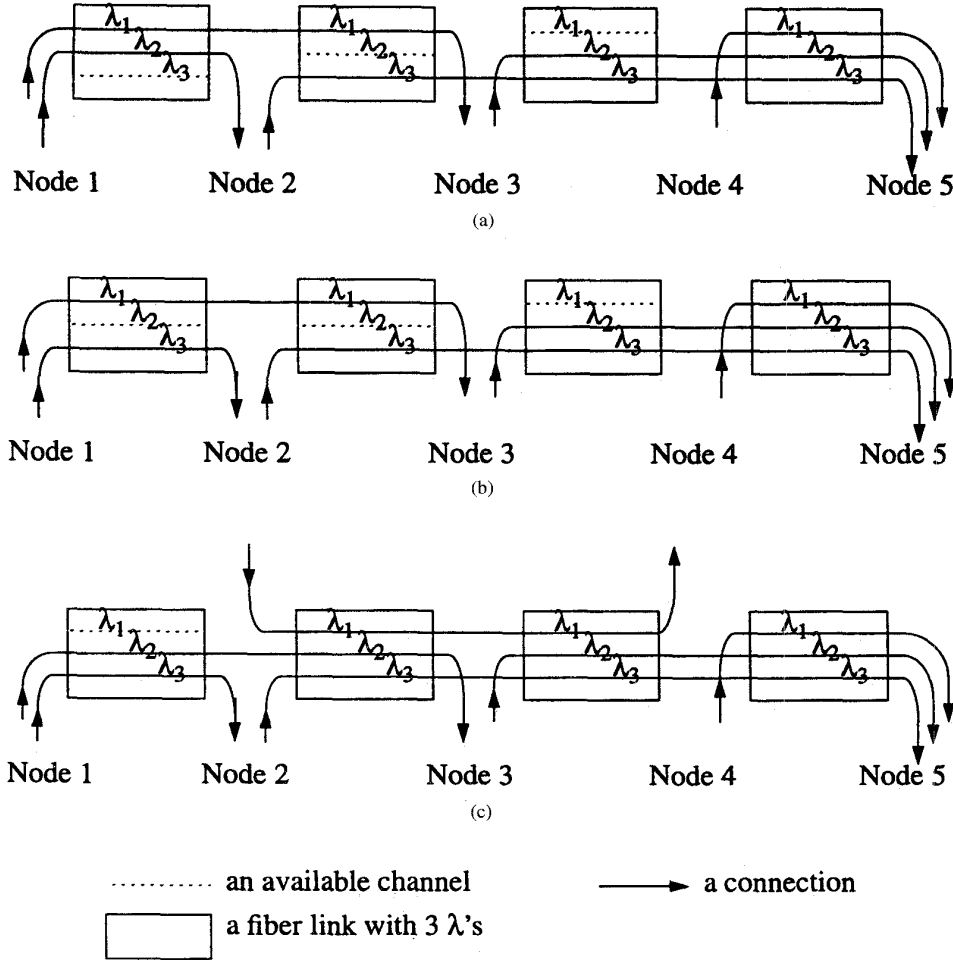


Fig. 6. An example which parallel MTV_WR rerouting cannot solve. The circuits should be rerouted in sequence (i.e., sequential MTV_WR rerouting).

the wavelength λ_1 is released to accept $u_{2 \rightarrow 4}$. Each of these rerouted circuits can be rerouted using MTV_WR operation. But since they do not occupy disjoint sets of links, they cannot be executed in parallel, i.e., they have to be rerouted sequentially. We call this sequential MTV_WR rerouting. This will increase the delay of connecting the new call.

In addition, in Fig. 7, the existing circuits $u_{1 \rightarrow 2}$ and $u_{1 \rightarrow 3}$ swap their wavelengths λ_1 and λ_2 to accept $u_{2 \rightarrow 4}$ on λ_1 . These rerouted circuits cannot be rerouted using the MTV_WR operation since their rerouting are on the same path but no vacant wavelength is available to move them. We call this wavelength-retuning rerouting. As these circuits are rerouted, they must be shut down, which incurs a very long disruption period, as in RAS rerouting. The parallel MTV_WR rerouting scheme introduces a relatively simpler implementation, shorter delay and disruption period, and less complexity than other rerouting algorithms. In the Appendix, we show that WR rerouting is an NP-complete or NP-hard problem. From the simulation, we will also show that the parallel MTV_WR rerouting scheme can achieve a performance close to the lower bound. So, we only focus on parallel MTV_WR rerouting in this paper.

IV. REROUTING ALGORITHM

The objective of the rerouting algorithm is to minimize the weighted number of existing circuits to be rerouted. Finding the optimal solution with the parallel MTV_WR rerouting scheme can be solved in polynomial time as opposed to that of a general rerouting problem which is NP-complete or NP-hard.

A. Parallel MTV_WR Rerouting Algorithm

A few notations are defined as follows. Assume that an existing circuit $u \in U$ passes through the sequence of directed links $[i_j(u), i_{j+1}(u)]$, $j = 1, \dots, h(u)$, on the wavelength $\lambda(u)$ (i.e., the directed edges, $[i_j^{\lambda(u)}(u), i_{j+1}^{\lambda(u)}(u)]$, on the graph in Fig. 2). Define $g(u)$, $u \in U$, as the retuning variable: $g(u) = \lambda$ if the circuit u can be retuned to the vacant wavelength on the same path with the smallest index λ , i.e., $\lambda \triangleq \min \{\lambda' \in \Lambda : q[i_j^{\lambda'}(u), i_{j+1}^{\lambda'}(u)] = NULL, j = 1, \dots, h(u)\}$, and $g(u) = NULL$, otherwise. A retunable circuit u is $g(u) \neq NULL$.

The algorithm transforms the network to a graph, labels the edges of the graph with appropriate cost, and searches for the best rerouting solution to route the new connection request.

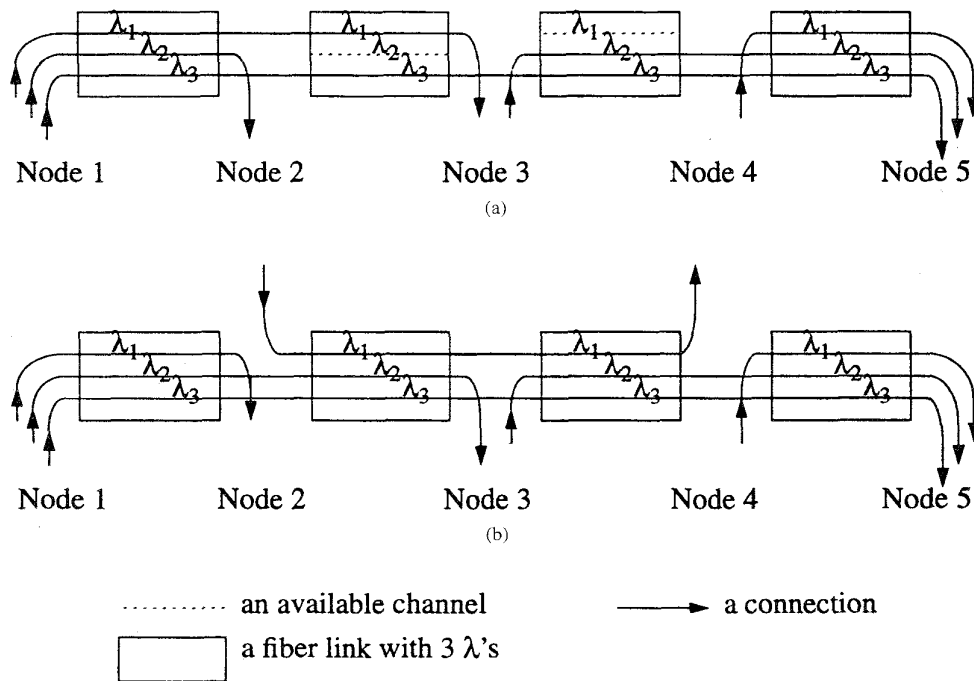


Fig. 7. An example which parallel MTV-WR rerouting cannot solve. The rerouted circuits cannot be moved to a vacant wavelength while being on the same path (i.e., wavelength-retuning rerouting).

1) *Graph Transformation*: A graph similar to that used in phase-I routing $\cup_{\lambda \in \Lambda} G(N^\lambda, L^\lambda)$ is created by adding some crossover edges along the route of a retunable circuit, i.e., $[i_j^{\lambda(u)}(u), i_k^{\lambda(u)}(u)]_u, \forall j, k, h(u)+1 \geq k > j \geq 1, k-j \geq 2$ where $g(u) \neq NULL$. Notice that $(*, *)|_u$ is used because more than one retunable circuit may connect the same pair of nodes on the same wavelength, and therefore more than one crossover edge may connect the same pair of vertices, each for a retunable circuit. These crossover edges are used to count the weighted number of rerouted circuits to accommodate the new connection request. Thus, the new graph is $\cup_{\lambda \in \Lambda} G(V^\lambda, E^\lambda)$ where

$$\begin{aligned}
 V^\lambda &= N^\lambda, \lambda \in \Lambda, \\
 E^\lambda &= L^\lambda \cup \{[i_j^{\lambda(u)}(u), i_k^{\lambda(u)}(u)]_u : \\
 &\quad h(u) + 1 \geq k > j \geq 1, k - j \geq 2, \\
 &\quad g(u) \neq NULL, \lambda(u) = \lambda, u \in U\}.
 \end{aligned}$$

A typical example is shown in Fig. 8.

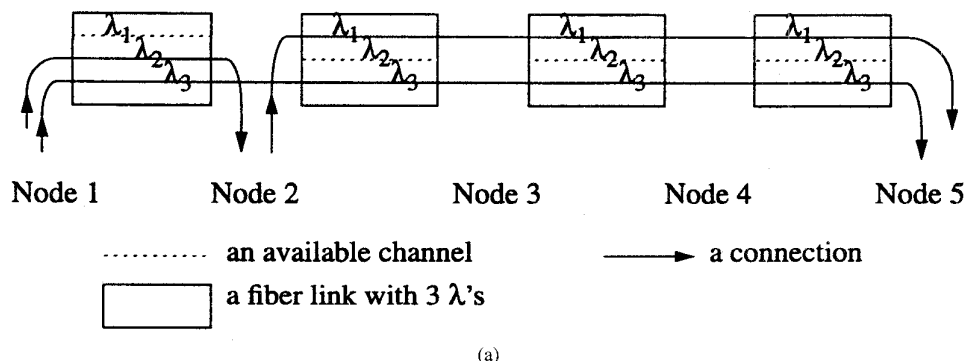
A retunable edge is either a crossover edge or an edge associated with a busy channel used by a retunable circuit (namely, in the set of $\{(i^\lambda, j^\lambda) : (i, j) \in L, q(i^\lambda, j^\lambda) \neq NULL, g[q(i^\lambda, j^\lambda)] \neq NULL\}$). Idle edges are in the set of $\{(i^\lambda, j^\lambda) : (i, j) \in L, q(i^\lambda, j^\lambda) = NULL\}$. The nonretunable edges are those in the set of $\{(i^\lambda, j^\lambda) : (i, j) \in L, q(i^\lambda, j^\lambda) \neq NULL, g[q(i^\lambda, j^\lambda)] = NULL\}$. We define a function $\tilde{q}(e) = u$ for edge e on the auxiliary graph if retunable edge e is associated with a retunable circuit u . $\tilde{q}(e) = NULL$ if e is an idle edge, and $\tilde{q}(e) = \infty$ if e is a nonretunable edge.

2) *Cost Labeling*: Cost for an idle edge is a tiny positive constant ϵ while cost for a nonretunable edge is infinite. Cost for a retunable edge associated with retunable circuit u is c_u which is a positive weighting factor indicating the penalty of rerouting an existing circuit u to accommodate the new connection. $\min_{u \in U} c_u > |L|\epsilon$ is assumed to prevent our objective of minimization of the weighted number of rerouted circuits from being changed by large values of ϵ . Actually, ϵ thus chosen will force the given optimization to minimize the weighted number of rerouted circuits and then minimize the number of idle channels to place the new circuit for the same minimal weighted number of rerouted circuits.

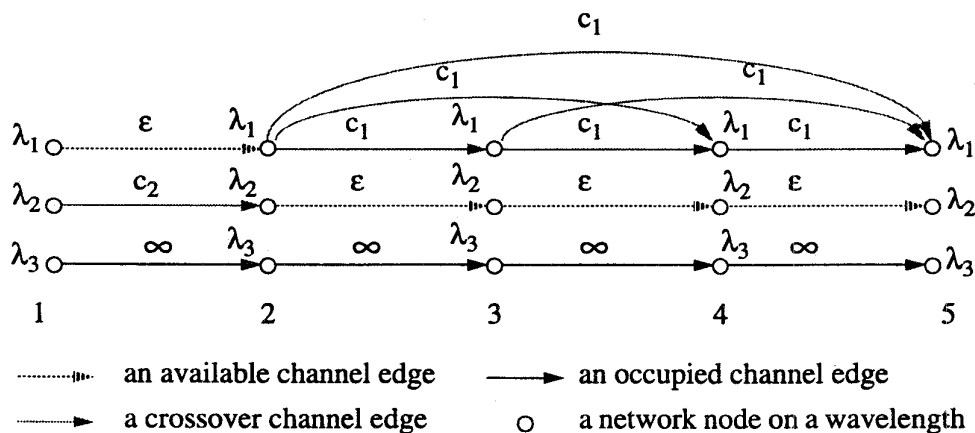
3) *Route Searching*: Similar to phase I of the routing algorithm, the least costly wavelength and associated shortest path will be sought for serving the new connection request. If the resulting minimum cost is finite, the new connection can be successfully accommodated using the obtained least costly route as follows.

- If this route traverses an idle edge, the new circuit should be routed through the corresponding idle channel.
- If this route traverses a retunable edge of a retunable circuit, the new circuit should be routed through those channel edges which are on this retunable circuit and between the starting vertex and the ending vertex of this edge. (For example, in Fig. 8, the crossover edge from nodes 2–4 on wavelength λ_1 of the retunable circuit $u_{2 \rightarrow 5}$ indicates routing using channels (2, 3) and (3, 4) on wavelength λ_1 .)

Those existing circuits with any of their channel overlapping with this new circuit should be rerouted to the smallest index



(a)



(b)

Fig. 8. Illustration of the graph transformation of the parallel MTV_WR rerouting algorithm.

of vacant wavelength on the same path as indicated by the retuning variable $g(u)$.

On the other hand, if the resulting cost of route searching is infinite, the new connection request should be rejected.

In the Appendix, we prove the correctness of this algorithm which is a polynomial-time algorithm. The computation includes creating the auxiliary graph and finding the shortest paths on the graph. The computation of the graph transformation is bounded by the calculation of the status functions, i.e., $O(|U||L||\Lambda|)$, and creating the channel and crossover edges, i.e., $O(|L||\Lambda|)$ and $O(|U||N|^2)$, respectively. Using Dijkstra's algorithm to find the shortest path on each subgraph with a total number of $|N|$ vertices requires $O(|N|^2)$ since the graph transformation does not create new vertices other than those in N although crossover edges are created. Therefore, finding the shortest paths needs complexity $O(|N|^2|\Lambda|)$, and selection of the minimum cost route needs complexity $O(|\Lambda|)$. Then the overall algorithm has complexity $O(|U||N|^2|\Lambda|)$.

V. PERFORMANCE RESULTS

We use the 21 node, 26 bidirectional-link ARPA2 network as the topology to generate the numerical results. Input traffic is generated uniformly from each of 21 nodes and is destined to each of the other 20 nodes with equal probability. The con-

nection requests of each node arrive as independent Poisson processes with common arrival rate r and hold the circuits with independent exponential times with common mean holding time equal to 1. Each node has enough transmitters and receivers such that the connect requests can only be blocked due to a lack of bandwidth or due to conflicts with the wavelengths of other existing circuits. A blocked connection request is cleared and will not retry.

Fig. 9 plots blocking probability versus arrival rate per node. The uppermost curve is the wavelength-continuous network. The lowest is the wavelength-shifting network in which a circuit can change its wavelength, thereby relaxing the wavelength continuity constraint. The performance of the wavelength-shifting network [8] is used as a lower bound to the wavelength-continuous network with rerouting because rerouting can alleviate the wavelength continuity constraint but cannot eradicate it. The middle curves are the wavelength-continuous networks with the parallel MTV_WR rerouting scheme. Two kinds of rerouting cost are assumed: equal weight, i.e., $c_u = c$, $u \in U$, and linear weight which is proportional to the number of hops in the rerouted circuits, i.e., $c_u = h(u)$, $u \in U$. The middle curves show that the equal weight scheme has smaller blocking probability than the linear weight scheme. The curves show that rerouting can improve blocking probability while the performance is close

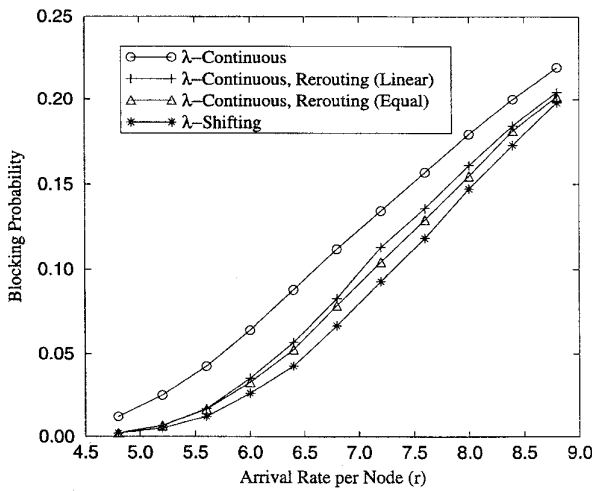


Fig. 9. Blocking probabilities of a wavelength-continuous network, a wavelength-continuous network with rerouting, including equal weight and linear weight, and a wavelength-shifting network versus traffic loading per node, r .

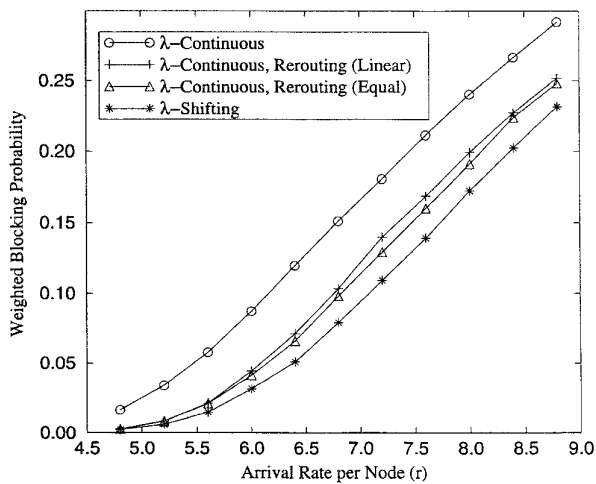


Fig. 10. Weighted blocking probabilities of a wavelength-continuous network, a wavelength-continuous network with rerouting, including equal weight and the linear weight, and a wavelength-shifting network versus traffic loading per node, r .

to the lower bound. On the average, call blocking probability is reduced by 30% with rerouting.

Fig. 10 compares the weighted blocking probabilities of wavelength-continuous networks with and without rerouting. The weight of a circuit is the number of shortest hops between the origin and the destination. Weighted blocking probability reflects the fact that blocking a long-hop connection costs more than blocking a short-hop connection in a real situation. Fig. 10 still shows that the weighted blocking probability can be improved with rerouting. In addition, the performance gain in weighted blocking is larger than that shown in Fig. 9 especially for large network input traffic. This implies that rerouting can effectively accommodate long-distance call requests in network congestion.

Fig. 11 plots the average number of rerouted circuits each time the parallel MTV-WR rerouting scheme is executed.

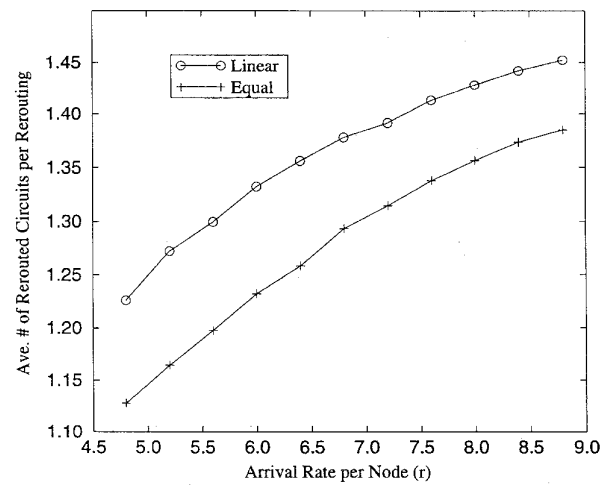


Fig. 11. Average number of rerouted circuits each time at the execution of rerouting versus traffic loading per node, r , for equal weight and linear weight of rerouting.

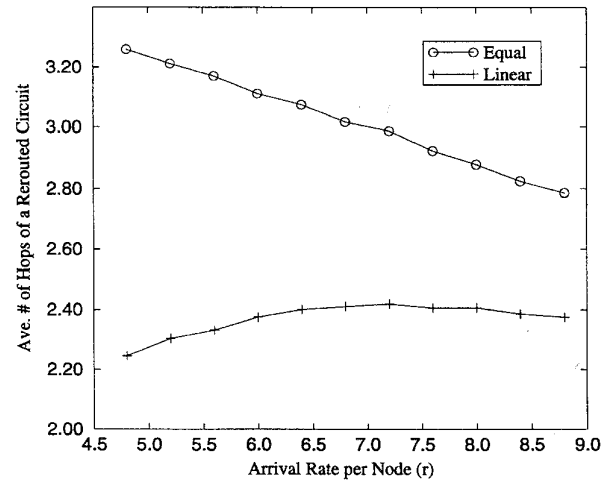


Fig. 12. Average number of hops of the rerouted circuits versus traffic loading per node, r , for equal weight and linear weight of rerouting.

The curves show that the average number of rerouted circuits increases as the traffic loading increases. The equal weight scheme reroutes fewer circuits than does the linear weight scheme because we use equal weight to minimize the number of circuits rerouted. The actual number of circuits rerouted is very small (<2). For the equal weight scheme, it is about 1.3 on the average. In addition, the disruption period of existing circuits is very short ($\approx \mu\text{s}$), and therefore the incurred disruption of the proposed rerouting should be little.

Fig. 12 plots the average number of hops of rerouted circuits versus the arrival rate per node. The curves show that the linear weight scheme will tend to reroute shorter-hop circuits than the equal weight one. For the equal weight scheme, the average number of hops of rerouted circuits decreases when traffic loading increases.

From Figs. 9–12, we conclude that the equal weight performs better than the linear weight scheme in our example.

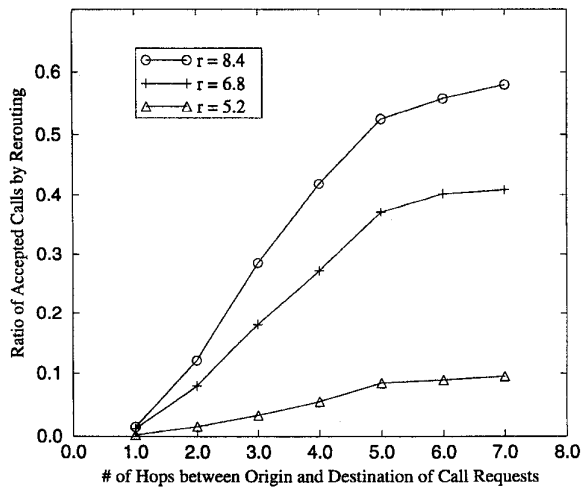


Fig. 13. Ratio of the number of accepted calls by equal-weight rerouting over the number of accepted new calls versus the number of hops between the origin and destination of new call requests at different traffic loading per node.

Fig. 13 plots the ratio of accepted new calls by equal-weight rerouting over the total number of accepted new calls versus the number of shortest hops between the origin and the destination of the new call requests. The curves show that longer-hop new calls need more rerouting than the shorter-hop new calls.

VI. CONCLUSION

In this paper, we employ rerouting to improve the call blocking probability due to the wavelength continuity constraint in circuit-switched wavelength-division-multiplexed all-optical networks. To minimize disruptions of the existing connections, move-to-vacant wavelength-retuning (MTV_WR) is proposed as the circuit migration scheme whereby the disruption period can be reduced by two orders of magnitude compared to the reroute-after-shutdown rerouting scheme in a wide-area network. parallel MTV_WR routing is used to reroute multiple circuits on disjoint sets of links. An optimal rerouting algorithm is developed to select a minimum number of rerouted circuits to accommodate a new connection with the parallel MTV_WR routing. Performance results show that call blocking probability is reduced by an average of 30%. In addition, each rerouting only reroutes an average of 1.3 circuits in a 21-node test network.

APPENDIX

In the following, we will show that the decision problem of the wavelength-retuning rerouting problem is NP-complete. We will also show that the presented algorithm to solve the parallel MTV_WR rerouting problem is correct.

We use the NP-completeness property of the static lightpath establishment problem (SLE) in [5] to show that the rerouting problem in general is computationally intractable.

Definition 1: (SLE) Given a WDM wavelength-continuous circuit-switched network $G(N, L)$ as assumed before, a set of wavelengths Λ , a set of circuits $U = \{u_1, u_2, \dots, u_n\}$ with

predetermined paths, is it possible to place all these circuits in the network by properly assigning wavelengths to them?

The following decision problem related to the wavelength-retuning rerouting is used to show that the optimization problem of the rerouting problem in general is very difficult.

Definition 2: (WRR) Given a WDM wavelength-continuous, circuit-switched network $G(N, L)$, a set of wavelengths Λ , a set of existing circuits $U = \{u_1, u_2, \dots, u_n\}$, and a new connection request u_{n+1} with predetermined path, is it possible to place this new request in the network without rerouting or with rerouting by only retuning the wavelengths of certain circuits in U ?

Theorem 1: (WRR) is NP-complete.

Proof: (WRR) can be polynomially transformed to (SLE) by including the new connection request and the existing circuits in a single set of connections and disregarding the wavelengths that the existing circuits used to be placed. If the solution of (SLE) is available, it can be interpreted as that of (WRR) by checking the wavelength of the new connection and whether or not the wavelengths of existing connections have been changed. If (SLE) cannot find a solution, (WRR) should not have any solution since the (SLE) is more flexible than (WRR). So (WRR) can be polynomially reduced to (SLE).

On the other hand, (SLE) can be transformed to a sequence of subproblems (WRR_i) , $i = 1, 2, \dots, n$. Then (WRR_i) is defined as given the set of the placed circuits $U_{i-1} = \{u_1, u_2, \dots, u_{i-1}\}$ ($U_0 = \phi$), is it possible to place the connection u_i with predetermined path without rerouting or with rerouting by only retuning the wavelengths of connections in U_{i-1} ? If the solution of (WRR_n) is available, it should be the solution of (SLE). If any of (WRR_i) , $i = 1, 2, \dots, n$ cannot find a solution, then (SLE) cannot find a solution; otherwise if (SLE) has a solution and (WRR_j) does not have a solution, then there will be a contradiction because the routes of connections u_1, u_2, \dots, u_j obtained in (SLE) can be used as the solution for (WRR_j) . So, the computational complexity of (SLE) should be at most of the order of $O(n)O(WRR_n)$. Therefore, (SLE) can be polynomially reduced to (WRR).

Hence, (WRR) is NP-complete. \square

We then show that the presented algorithm to solve the parallel MTV_WR rerouting problem is correct.

Definition 3: (P MTV_WR) Given a WDM wavelength-continuous, circuit-switched network $G(N, L)$ as assumed, a set of wavelengths Λ , a set of existing circuits $U = \{u_1, u_2, \dots, u_n\}$ and their associated routes, and a blocked new connection request u_0 , is it possible to place this new request in the network using the parallel MTV_WR rerouting scheme while minimizing the weighted number of existing connections to be rerouted?

Theorem 2: The presented algorithm can solve (P MTV_WR).

Proof: We consider the original network and define the following terminologies. A route p^λ of the original network is denoted by the sequence of channels $(i_1^\lambda, i_2^\lambda, \dots, i_h^\lambda, i_{h+1}^\lambda)$. A route is retunable if $u = q(i_j^\lambda, i_{j+1}^\lambda)$, $u \neq NULL$, implies $g(u) \neq NULL$, $j = 1, \dots, h$. The cost of a retunable route is the weighted number of retunable circuits plus the product of a tiny positive constant ϵ and the number of idle channels on it.

Notice that although ϵ is used, it will not change the objective of minimization of the weighted number of rerouted circuits as mentioned in Section IV. A retunable route is *noninterleaved* if given there exist j and k ($1 \leq j \leq k \leq h$) such that $q(i_j^\lambda, i_{j+1}^\lambda) = q(i_k^\lambda, i_{k+1}^\lambda)$ and $q(i_j^\lambda, i_{j+1}^\lambda) \neq NULL$, then $q(i_j^\lambda, i_{j+1}^\lambda) = q(i_t^\lambda, i_{t+1}^\lambda), \forall t, 1 \leq j \leq t \leq k \leq h$; otherwise, it is an interleaved retunable route. This means that if one traverses a noninterleaved route, one will see a particular circuit continuously rather than in discontinuous sets of channels.

Lemma 1: For any retunable route p^λ connecting u_0 , there always exists a noninterleaved retunable route p'^λ connecting this new connection with the same or less cost than p^λ on the original network.

Proof: Whenever there exists an index t such that $q(i_j^\lambda, i_{j+1}^\lambda) = q(i_k^\lambda, i_{k+1}^\lambda)$ and $q(i_j^\lambda, i_{j+1}^\lambda) \neq NULL$, but $q(i_j^\lambda, i_{j+1}^\lambda) \neq q(i_t^\lambda, i_{t+1}^\lambda)$ ($1 \leq j < t < k \leq h$), we can always delete all the channels $(i_{j+1}^\lambda, i_{j+2}^\lambda), \dots, (i_{k-1}^\lambda, i_k^\lambda)$ from p^λ and add some additional channel(s) traversed by the circuit $u = q(i_j^\lambda, i_{j+1}^\lambda)$ from i_{j+1}^λ to i_k^λ . This delete-and-add operation should possibly result in eliminating a retunable circuit or an idle channel, thereby reducing cost or maintaining the same cost. This procedure will stop when a noninterleaved route p'^λ is obtained. The obtained noninterleaved retunable route can connect the new connect request (the starting and the ending points will never be deleted) with the same or less cost.

Lemma 1 shows that we only need to consider the noninterleaved retunable routes on the original network to find the optimal solution. In reality, the optimal solution in our cost objective (minimization of the weighted number of rerouted circuits) can be chosen from the least costly one from all the noninterleaved retunable routes connecting the new connection request u_0 .

Consider the phase-II auxiliary graph. A route p^λ is denoted by the sequence of edges e_1, e_2, \dots, e_h . A feasible route p^λ on the auxiliary graph is the one with each edge either an idle edge or a retunable edge associated with a retunable circuit. A *single* feasible route on the auxiliary graph is defined as one on which at most one edge on this route is associated with the same retunable circuit. That is, if there exists e_j such that $\tilde{q}(e_j) = u \in U$, then $\tilde{q}(e_t) \neq u, \forall t, 1 \leq t \leq h$ and $t \neq j$; otherwise, it is a nonsingle feasible route. For example, in Fig. 8, the crossover edge connecting nodes 2–4 on λ_1 is a single feasible route while the route consisting of the edges from nodes 2–3 and from nodes 3–4 is not. The interesting property is that a nonsingle feasible route has higher cost than its associated single feasible one.

Lemma 2: For any nonsingle feasible route p^λ connecting u_0 , there always exists a single feasible route p'^λ connecting u_0 with strictly less cost.

Proof: p^λ should have more than one retunable edge associated with a particular retunable circuit because the new connection is blocked and it is nonsingle. Assume that two edges e_j and e_k are associated with the same retunable circuit where $k > j$. Then we can create a new route by replacing all edges $e_t, j \leq t \leq k$ by only one crossover edge $[v_s(e_j), v_e(e_k)]$ where $v_s(e)$ and $v_e(e)$ denote the starting and

the ending vertices of edge e , respectively. The delete-and-add operation will reduce at least one edge with nonzero cost. This whole procedure will stop when a single feasible route p'^λ is found, resulting in strictly less cost.

Lemma 2 shows that the shortest path algorithm applied on the phase-II auxiliary graph will generate a single feasible route if the problem is feasible.

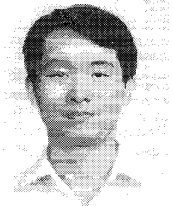
We can establish a one-to-one correspondence between the set of noninterleaved retunable routes on the original network, denoted by S_n , and the set of single feasible routes on the phase-II auxiliary graph, denoted by S_s , by the one-to-one and onto mapping $\mathcal{F}: S_n \rightarrow S_s$ as follows: i) if an idle channel on the original network lies along a noninterleaved route, then \mathcal{F} will create an idle edge corresponding to that channel; ii) if a retunable circuit overlaps with a noninterleaved route, \mathcal{F} will create one retunable edge connecting the first vertex to the last vertex where this retunable circuit overlaps with this route (e.g., in Fig. 8 a crossover edge connecting nodes 2–4 is created if a route traverses retunable circuit $u_{2 \rightarrow 5}$ only on the two channels from nodes 2–3 and from nodes 3–4 of wavelength λ_1). \mathcal{F} is onto: for any noninterleaved retunable route with certain retunable circuit lying continuously along this route on the original network, only one retunable edge associated with this circuit on the auxiliary graph is created, namely, the necessary condition for a single feasible route. The one-to-one property of \mathcal{F} can be shown by the result that only one unique noninterleaved retunable route is associated with a single feasible route. This result can be verified from edge to edge. For an idle edge on this single feasible route, only one unique idle channel can be associated with it. For a retunable edge on this single feasible route, only one unique retunable circuit, and therefore one unique sequence of channels between the starting and the ending vertices of this edge and on this retunable circuit can be associated with it.

It remains to show that the shortest route on the auxiliary graph leads to our optimal solution. Notice that the cost value of a single feasible route in S_s is the same as that of the corresponding noninterleaved retunable route under \mathcal{F} in S_n . However, the optimal solution of the shortest path algorithm on the auxiliary graph is in S_s while Lemma 1 shows that the real optimal solution of the original network is in S_n . Therefore, from the one-to-one correspondence between S_n and S_s and the identical cost value of two corresponding routes, we conclude that the shortest path algorithm on the auxiliary graph leads to the optimal solution of our rerouting problem. \square

REFERENCES

- [1] P. E. Green, Jr., *Fiber Optic Networks*. Englewood Cliffs, NJ: Prentice-Hall, 1993.
- [2] B. Mukherjee, "WDM-based local lightwave networks—Part I: Single-hop systems," *IEEE Network Mag.*, pp. 12–27, May 1992.
- [3] ———, "WDM-based local lightwave networks—Part II: Multihop systems," *IEEE Network Mag.*, pp. 20–31, July 1992.
- [4] R. Ramaswami, "Multiwavelength lightwave networks for computer communication," *IEEE Commun. Mag.*, pp. 78–88, Feb. 1993.
- [5] I. Chlamtac, A. Ganz, and G. Karmi, "Lightpath communications: An approach to high bandwidth optical WAN's," *IEEE Trans. Commun.*, vol. 40, pp. 1171–1182, July 1992.

- [6] T. E. Stern, K. Bala, S. Jong, and J. Sharony, "Linear lightwave networks: Performance issues," *J. Lightwave Technol.*, vol. 11, pp. 936-950, May/June 1993.
- [7] J.-F. P. Labourdette, G. W. Hart, and A. S. Acampora, "Branch-exchange sequences for reconfiguration in lightwave networks," *IEEE Trans. Commun.*, vol. 42, pp. 2822-2832, Oct. 1994.
- [8] K.-C. Lee and V. O. K. Li, "A wavelength-convertible optical network," *J. Lightwave Technol.*, vol. 11, pp. 962-970, May/June 1993.



Kuo-Chun Lee (S'92-M'95) was born in Taipei, Taiwan, Republic of China. He received the B.S. and M.S. degrees in electrical engineering from the National Taiwan University in 1982 and 1984, respectively, and the Ph.D. degree in electrical engineering from the University of Southern California in 1994.

From 1984 to 1986, he served as Maintenance Officer of Military Integrated Communication System, Ministry of Defense, Republic of China. In 1986, he was a Lecturer at the Kwang-Wu Junior College of Technology, Taipei, Taiwan. In 1987, he was a Research Assistant at the Institute of Mathematics, Academia Sinica, Republic of China. In 1995, he joined the Qualcomm Incorporated, San Diego, CA, where he has been engaged in the design and development of CDMA mobile station controllers. His research interests include all-optical networks, high-speed networks, and wireless communication networks.



Victor O. K. Li (S'80-M'81-SM'86-F'92) was born in Hong Kong in 1954. He received the S.B., S.M., and Sc.D. degrees in electrical engineering and computer science from the Massachusetts Institute of Technology, Cambridge, MA, in 1977, 1979, and 1981, respectively.

Since February 1981, he has been with the University of Southern California (USC), Los Angeles, CA, where he is Professor of Electrical Engineering and Director of the USC Communication Sciences Institute. His research interests include high-speed communication networks, personal communication networks, multimedia systems, distributed databases, queueing theory, graph theory, and applied probability. He has lectured and consulted extensively around the world.

Dr. Li is very active in the Institute of Electrical and Electronic Engineers (IEEE), having been a member of the Computer Communications Technical Committee since 1983, and having served as Chairman from 1987 to 1989. He served as Chairman of the Los Angeles Chapter of the IEEE Information Theory Group from 1983 to 1985. He is the Steering Committee Chair of the International Conference on Computer Communications and Networks (IC3N), General Chair of the 1st Annual IC3N, June 1992, Technical Program Chair of the Institution of Electrical Engineers (IIE) Personal Communication Services Symposium, June 1995, General Chair and Technical Program Chair of the 4th IEEE Workshop on Computer Communications, October 1989. A member of ACM and ORSA, Dr. Li has served as an Editor of IEEE NETWORK and of *Telecommunication Systems*, Guest Editor of IEEE JOURNAL ON SELECTED AREAS COMMUNICATIONS and of *Computer Networks and ISDN Systems*, and is now serving as an Editor of *ACM Wireless Networks*. He has served on the International Advisory Board of IEEE TENCON'90, IEEE TENCON'94, IEEE SICON'91, IEEE SICON'93, IEEE SICON/ICIE'95, the International Conference on Microwaves and Communications '92, and the International Symposium on Communications '91. He is an IAE Fellow, a New York City Urban Fellow, and is listed in *Marquis' Who's Who in Frontier Science and Technology*, *Who's Who in California*, and *Who's Who Among Asian Americans*.