Minimizing Wavelengths in an All-Optical Ring Network

Gordon Wilfong

Bell Laboratories, Murray Hill NJ 07974, USA

Abstract. In an optical network, wavelength-division multiplexing permits multiple messages to be sent along the same link in the network by transmitting each message using a distinct wavelength. In such a network, the action of establishing a connection between two nodes entails choosing a route in the network between the nodes and then assigning to the connection, a wavelength for each link in the chosen route. We will consider problems concerning the minimum number of wavelengths sufficient to establish a set of connections. In a wavelength-selective (WS) network, the assigned wavelengths for any given connection must be the same throughout the chosen route. In contrast, for a wavelength-interchanging (WI) network the assigned wavelengths for each link in the route may differ. We show that for an important class of networks, namely ring networks of n nodes with clockwise and counterclockwise links between each neighboring pair of nodes on the ring, the minimum number, k_n, of wavelengths that suffice to allow simultaneously all possible connections is the same for either case. In particular, for both WS and WI networks, it is shown that $k_n = (n^2-1)/8$ if n is odd, $k_n = n^2/8$ if n is even and divisible by 4 and $k_n = n^2/8 + 1/2$ if n is even but not divisible by 4.

1 Introduction

Wavelength-division multiplexing appears to be an important and effective solution for increasing the bandwidth within an optical fiber [6]. However, while multiwavelength transmission allows for increased bandwidth along optical fibers, the processing, switching and buffering delays at the nodes can result in bottlenecks. A solution to this problem was offered by the idea of "wavelength routing" [3, 10]. That is, the wavelength of the transmission, used without further processing, determines the routing of the signal. Such networks are called "all-optical networks" [2] and they require circuit-switching. Thus for a given connection between two nodes to be achieved, a route through the network must be determined and a wavelength allocated to the connection for each link along the route. The resulting route and allocation of wavelengths is referred to as an "optical path" [9].

There are two general types of networks depending on how optical paths are assigned. In a wavelength-selective (WS) network [4], an optical path for a given connection is formed by assigning the same wavelength along each link in the chosen route. In a wavelength-interchanging (WI) network [7, 8], the wavelength of the transmission is chosen independently along each link. The advantage of a WS network over a WI network is that there is no need to perform wavelength conversions at each node along the route. However a key advantage to a WI network over a WS network is that it could potentially allow for the establishment of an optical path that would be impossible with a WS network. The simplicity of a WS network therefore could result in lower bandwidth or equivalently could require more capacity in terms of needing additional wavelengths to accommodate desired connections.

Wavelength assignments in WS networks have been referred to as "light paths" [4] but we will call them *monochromatic assignments* to emphasize the fact that the wavelength ("color" of the transmitting light) along each link is the same. It has been shown [4] that given a set of desired connections along with some fixed routing for each such connection the question as to whether it is possible to achieve monochromatic assignments along each route for a fixed number of available wavelengths is NP-complete. The proof uses a reduction from Graph Coloring [5]. Thus in this general case, even if the number of wavelengths is sufficient to support connections in a WI network it is difficult to determine whether or not a solution exists using monochromatic assignments. However, in the case where the network topology is a tree, the number of wavelengths required for all monochromatic assignments is the same as if general optical paths are permitted [4]. In the Open Problems section of [1] it is stated that "... (an) interesting and practically useful question is getting a good bound on the number of wavelengths required for a given network and a given message pattern". In this paper we show just such a result for an optical network that consists of a ring of n nodes such that between each pair of neighboring nodes on the ring there are two links, one directed clockwise and the other directed counterclockwise and the set of connections to be established consists of a connection between each pair of nodes in the network. For such a network we determine k_n, the minimum number of wavelengths that allow for the establishment of general (i.e. not necessarily monochromatic) optical paths for connections between every pair of nodes. This computation is constructive in the sense that a method for routing each connection is given that allows for optical paths for all connections and uses at most k_n wavelengths. Then we show that even if monochromatic assignments are required, k_n wavelengths remain sufficient to establish optical paths for all connections. Again, this is done constructively so that not only is the method for choosing routes for each connection given but also an algorithm for choosing the wavelengths assigned to each route is given and proven correct. These results can be compared to the general results in [1] that imply that for the case under consideration, $O(\sqrt{n}k_n)$ wavelengths are sufficient.

2 Definitions and Problem Descriptions

Consider a ring network $\mathcal{R}_n$ consisting of n nodes $v_0, v_1, \ldots, v_{n-1}$ in clockwise order. We assume that n, the size of the network, is fixed and so we use the notation $i \oplus j$ to denote $i + j \pmod n$ and similarly $i \ominus j$ is used to mean

$i - j \pmod n$. For each i, $0 \le i \le n-1$, there is a clockwise link, denoted by α_i, from node v_i to node $v_{i\oplus 1}$. Similarly, there is a counterclockwise link, denoted by β_i from node v_i to node $v_{i\ominus 1}$. Establishing a connection from one node v_i to another v_j, that is, determining an optical path from v_i to v_j, entails choosing a route from v_i to v_j and assigning a wavelength along each link in the route. Thus in order to be able to establish a given set of such connections, there needs to be an available wavelength along each link for each route traversing the link. This implies there is a lower bound on the number of wavelengths that would allow all such connections to be established. This work is concerned with the problem of determining such lower bounds and then showing how to route connections and assign wavelengths along the routes using the minimum number of wavelengths. These problems are considered for both WI and WS networks.

Each link will be assumed to have the same number of wavelengths and we refer to each wavelength along a link as a *channel* of the link. That is, each connection will require one channel along each link in its route and each link is thought of as having some number of channels one of which is assigned to each connection passing through the link. Then the first problem of interest is determining the minimum number of channels each link needs in order to be capable of establishing connections between each pair of nodes. Each channel represents a wavelength (or "color" of light) and so we think of each channel as being uniquely labeled by one of the k colors $c_1, c_2, \ldots, c_k$. The second problem considered is to determine how many different colors are required so that each connection can be established using channels all of the same color.

More formally, the action of establishing a connection from *starting node* v_i to *destination node* v_j requires choosing a *route* $R_{i,j}$ where a route consists of clockwise links $\alpha_i, \alpha_{i\oplus 1}, \ldots, \alpha_{j\ominus 1}$ or counterclockwise links $\beta_i, \beta_{i\ominus 1}, \ldots, \beta_{j\oplus 1}$. The links in a route are said to be *traversed* by the route. The number of links traversed by a route is called the *length* of the route. Given a route for a connection we sometimes refer to the *length of the connection* and by this we mean the length of the route chosen for the connection. The second step in establishing a connection between nodes v_i and v_j is to assign a *free* channel from each of the links traversed by the route $R_{i,j}$ where a channel is said to be free if it has not been assigned to any other connection. Thus the problem of determining the minimum number of wavelengths sufficient to allow establishing all connections in a WI network can be stated as follows.

Problem 1 *Suppose we wish to be able to establish connections from each node to every other node at the same time. For a n-node ring $\mathcal{R}_n$, what is the minimum number k_n such that if each link has k_n channels then all required connections can be established. That is, what is the minimum number of channels for each link sufficient to guarantee that for each connection a route can be chosen and free channels can be assigned for each link of the chosen route.*

Now we consider the problem of determining the minimum number of wavelengths sufficient to allow establishing all connections in a WS network. That is, for each route $R_{i,j}$ all channels assigned to $R_{i,j}$ must not only be free but must all have the same color. Such an assignment of channels to a route is said to be *monochromatic.*

Problem 2 *Suppose we wish to be able to establish connections from each node to each other node at the same time. For a n-node ring $\mathcal{R}_n$, where each link has k_n channels, is it possible to choose routes for each required connection and to make monochromatic assignments of channels for each such route? That is, are k_n channels for each link sufficient to allow all assignments of channels to be chosen to be monochromatic?*

3 Minimum Number of Channels

We begin by considering **Problem 1**. That is, we wish to determine k_n, the minimum number of channels per link in $\mathcal{R}_n$ that guarantees that all required connections can be established in a WI network. An outline of the argument used to determine k_n is provided.

Suppose all connections are routed using routes of minimum length. Then no route is longer than $n/2$ for even n or $(n-1)/2$ for odd n. Notice that t_n, the total number of channels used by all the connections for this type of routing, is no more than would be used by any other routing strategy and $t_n = n(n^2-1)/4$ if n is odd and $t_n = n^3/4$ if n is even. Since t_n is the least number of links that can be traversed for any choice of routing, if a routing scheme results in having the routes evenly distributed so that each of the $2n$ links is traversed by exactly $t_n/2n$, the average number of routes per link, then this routing scheme is optimal in the sense that any other routing scheme will have some link with at least $t_n/2n$ routes traversing it. In other words, $k_n \geq t_n/2n$.

Thus in the case of odd n there would be $(n-1)(n+1)/8$ routes per link and for even n there would be $n^2/8$ routes per link. Notice that for odd n, since both $(n-1)$ and $(n+1)$ are divisible by 2 and one must be divisible by 4 the value of $(n-1)(n+1)$ is divisible by 8 and hence $(n-1)(n+1)/8$ is an integer. Therefore if we could show a method for routing the connections such that each link was traversed by exactly $(n-1)(n+1)/8$ routes then this would constructively show that $k_n = (n-1)(n+1)/8$. Similarly for even n where n is divisible by 4, the amount $t_n/2n = n^2/8$ is an integer and so our goal will be to describe a routing method that results in each link being traversed by exactly $n^2/8$ routes enabling us to conclude that $k_n = n^2/8$ in this case. For even n where n is not divisible by 4, the value $n^2/8$ is not an integer and so a routing technique to evenly distribute the routes over the links is not possible. However this says that for any routing there is at least one link that is traversed by at least $n^2/8 + 1/2$ routes. Therefore we wish to show that there is a routing method such that no link is traversed by more than $n^2/8 + 1/2$ and so conclude that in this case $k_n = n^2/8 + 1/2$.

It can be shown that when n is odd, the unambiguous routing strategy of choosing the shortest route for each connection results in each link being traversed by exactly $(n^2-1)/8$ routes. Since $(n^2-1)/8 = t_n/2n$, we conclude that $k_n = (n^2-1)/8$.

We then consider the case where n is even. As before we choose to route connections via a shortest route but in the case of routes of length $n/2$ this decision is ambiguous. This is corrected by choosing the clockwise route for each connection of length $n/2$ starting at an even numbered node (that is, those nodes v_i where i is even) and choosing the counterclockwise route for each connection of length $n/2$ starting at an odd numbered node. For the case where n is not only even but also divisible by 4 it can be shown that the routing strategy given results in each link being traversed by $n^2/8$ routes and hence $k_n = n^2/8$. In the case where n is even but not divisible by 4 it can be shown that the routing scheme results in no link being traversed by more than $n^2/8 + 1/2$ routes and so we conclude that in this case $k_n = n^2/8 + 1/2$.

Therefore collecting all the above results we can write the value of k_n, the least number of channels that permit all connections to be established simultaneously, as

$$k_n = \begin{cases} \frac{n^2-1}{8} & \text{if } n \text{ is odd} \\ \frac{n^2}{8} & \text{if } n \text{ is even and divisible by 4} \\ \frac{n^2}{8} + \frac{1}{2} & \text{if } n \text{ is even but not divisible by 4} \end{cases} \tag{1}$$

and it is the case that if all connections are routed via a shortest route (where in the case of even n the connections of length $n/2$ are routed clockwise if they start at an even numbered node and counterclockwise otherwise) then this results in each link being traversed by at most k_n routes thus achieving the lower bound.

4 Monochromatic Assignment of Channels to Routes

Next we consider **Problem 2** and in fact show that all required connections can be achieved using routes that allow for monochromatic assignments when only k_n colors are available. A proof outline is provided while most of the technical proofs are omitted in this abstract.

It will first be necessary to provide a routing strategy and then show how channels (colors) should be assigned in order to prove that monochromatic assignments are possible using k_n colors. As in the previous problem, we route all connections via shortest paths. We describe later how to resolve the question as to how to route the length $n/2$ connections when n is even. In any case, once the choice of routes has been fixed, the remaining problem is that of finding monochromatic assignments of channels for each route.

4.1 Monochromatic Assignments for Odd n

Since the shortest routes for connections in the case where n is odd are unambiguous we are left with only the problem of assigning colors to these routes. We consider the problem of assigning colors to the clockwise routes since the counterclockwise problem is independent and can be solved in an analogous fashion.

In this case, the results of the previous section show that the number of routes traversing channels of a given link, namely $k_n = (n^2 - 1)/8$, is the same for each link, and so the problem can be stated as follows. Partition the clockwise routes into k_n collections $S_1, S_2, \ldots, S_{k_n}$ so that for each i, $1 \leq i \leq k_n$, all routes in S_i are assigned the same color, say c_i and each S_i must be such that the set of routes in S_i together traverse every clockwise link exactly once. That is, for each α_i and each S_j there is exactly one route in S_j that traverses α_i.

Thus the problem of assigning colors to the routes can be summarized as follows for the case of odd n. For each color c_i, $1 \leq i \leq k_n$, find an ordered list of routes $S_i = (R_i(0), R_i(1), \ldots, R_i(m_i - 1))$ such that the destination vertex of $R_i(j)$ is the starting vertex of $R_i(j + 1 \pmod{m_i})$, $0 \leq j < m_i$, each clockwise link is traversed by exactly one $R_i(j)$ and each route appears in exactly one such list. We call such an ordered list of routes a *tour* since following each one in turn constitutes a non-overlapping traversal of the ring. Thus the S_i's partition the set of routes into k_n tours. Such an assignment of colors to routes (or routes to lists) is said to be a *valid assignment*. Therefore the goal is to find a valid assignment of colors to routes, thus proving that k_n colors is sufficient to achieve monochromatic assignments for all connections.

A convenient way to view this problem will be via a matrix representation as described next. Consider the matrix M with the entry in the i^{th} row and j^{th} column denoted by $M(i, j)$ where $0 \leq i < n$ and $1 \leq j \leq (n-1)/2$. Each $M(i, j)$ will be assigned a color c_t meaning that the route beginning at node v_i of length j is assigned color c_t or equivalently is placed in the list S_t.

```
proc color_odd(n) {     /*n is the number of nodes*/
    h := (n − 1)/2;
    r := 1;             /*c_r is the current color*/
    for(vertex = 0, 1, ..., h − 1) {
        for(column = h − vertex, h − vertex − 1, ..., 1) {
            M(vertex, column) := c_r;
            row := vertex + column;
            jump := h − vertex;
            while(row + jump < n) {
                M(row, jump) := c_r;
                row := row + jump;
            }
            jump := n − row + vertex;
            M(row, jump) := c_r;
            r := r + 1;
        }
    }
}
```

Fig. 1. Procedure *color_odd* for assigning colors to routes

In Fig. 1 we give a procedure, *color_odd*, written in pseudo-code that we claim assigns a color to every route and for any color, the set of resulting routes assigned that color forms a tour. Thus by the argument above, such a procedure is a constructive proof that k_n colors are sufficient for achieving monochromatic assignments.

We now outline the proof that the procedure *color_odd* satisfies

(i) the set of clockwise routes assigned a given color forms a tour and
(ii) every clockwise route is assigned exactly one color.

In the following discussions we will need to refer to the value of various variables in the procedure *color_odd* at various stages in the execution of *color_odd*. We use the notation $Val(var)$ to denote the value of variable var for the indicated stage in the execution. We begin by showing that the routes assigned any particular color form a tour.

Lemma 4.1 *The collection of routes assigned a given color by the procedure color_odd forms a tour.*

Proof: (Sketch) Notice that the color assigned to routes is fixed for fixed values of the variables *vertex* and *column* in the procedure. Consider a color c_r and let $Val(vertex) = i_r$ and $Val(column) = j_r$ be the values of variables *vertex* and *column* respectively during the execution of the procedure when routes are assigned the color c_r. Let $jmp = Val(h) - i_r$. Let $R_r(0), R_r(1), \ldots$ be the routes that are assigned color c_r where the routes are listed in the order in which they are assigned colors by the procedure. The starting vertex of $R_r(1)$ and the destination vertex of $R_r(0)$ clearly have index $i_r + j_r$. For $s = 1, 2, \ldots$, let $t_s = i_r + j_r + s * jmp$. A loop invariant argument shows that route $R_r(s)$ has starting node $v_{t_{s-1}}$ and its destination node is v_{t_s} for $s \geq 1$. The last route $R_r(m_r - 1)$ assigned color c_r occurs at line (13) and it can be shown that its starting node is the destination node of $R_r(m_r - 2)$ and has index given by the value of *row*.

Next we need to show that the destination node of $R_r(m_r - 1)$ is the starting node v_{i_r} of $R_r(0)$. Clearly, the index of the destination node of $R_r(m_r - 1)$ is given by the sum of the values of the variables *row* and *jump* which by the definition of *jump* is simply $n \oplus i_r = i_r$. Consider $Val(row)$, the value of the variable *row* at the time that $R_r(m_r - 1)$ is assigned color c_r. We must be somewhat careful here to make sure that a route of length $n - Val(row) + i_r$ is feasible. That is, it must be shown that $1 \leq n - Val(row) + i_r \leq (n-1)/2$. But the loop condition being false means exactly that $Val(row) + (n-1)/2 - i_r \geq n$ and so rewriting this gives $n - Val(row) + i_r \leq (n-1)/2$. If $Val(row) \geq n$ then the loop would have terminated before the previous iteration and so it must be that $Val(row) < n$ and hence $n - Val(row) + i_r > i_r \geq 1$. A monotonicity argument shows that no link is traversed by more than one route $R_r(s)$. ■

The last problem to complete the proof that *color_odd* produces a valid assignment is to show that every route is assigned exactly one color. For each i_p, $0 \leq i_p < (n-1)/2$, define the set $A(i_p)$ to be the set of all entries in M that are assigned a color in *color_odd* while the value of variable *vertex* is i_p.

We will show that each $A(i_p)$ consists of three regions of the matrix M, namely a horizontal strip $H(i_p)$, a vertical strip $V(i_p)$ and a diagonal strip $D(i_p)$ where

$$H(i_p) = \{M(i_p, j) : 1 \le j \le (n-1)/2 - i_p\},$$
$$V(i_p) = \{M(i, (n-1)/2 - i_p) : i_p < i \le (n-1)/2 + i_p\} \text{ and}$$
$$D(i_p) = \{M(i, j) : i + j = n + i_p, (n+1)/2 + i_p \le i \le n-1\}.$$

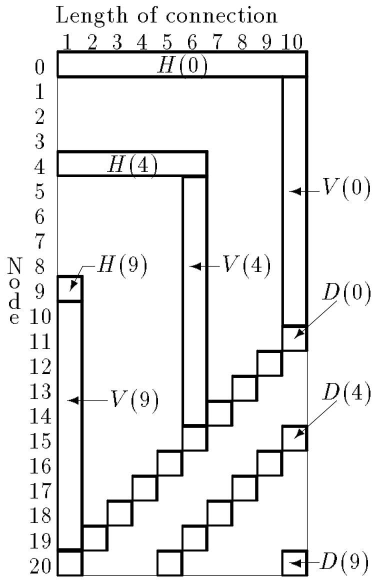

Fig. 2. Regions $H(i_p), V(i_p)$ and $D(i_p)$ for $i_p = 0, 4, 9$.

Given this characterization of $A(i_p)$, it can be shown that the $A(i_p)$'s form a partition of the elements of M and so our goal will be to show that for each fixed p, each element of $A(i_p)$ is assigned a color exactly once. Figure 2 shows some examples of the partition of the matrix M into regions $H(i_p), V(i_p)$ and $D(i_p)$ for various values of i_p where $n = 21$.

Throughout the remainder of this section, let i_p be some fixed value of the variable *vertex* where $0 \le i_p < (n-1)/2$. It is easy to check that $H(i_p)$, $V(i_p)$ and $D(i_p)$ form a partition of $A(i_p)$. Therefore it is sufficient to show that each element of $H(i_p)$, $V(i_p)$ and $D(i_p)$ is assigned a color exactly once by the procedure. It is straightforward to show that line (5) assigns a color to each element of $H(i_p)$ exactly once. Suppose we could prove the following relationships between the elements assigned colors by execution of line (9) of *color_odd*:

(i_V) the elements of M assigned colors in line (9) of the procedure are all elements of $V(i_p)$,

(ii_V) two distinct executions of line (9) of the procedure assign colors to distinct elements and

(iii_V) every element of $V(i_p)$ is assigned a color by some execution of line (9) of the procedure

Also, suppose we could prove analogous relationships (i_D), (ii_D) and (iii_D) between the elements assigned colors by execution of line (13) of *color_odd* and the elements of $D(i_p)$. Then we have shown that each element of $H(i_p)$, $V(i_p)$ and $D(i_p)$ is assigned a color exactly once by the procedure.

The following lemma (the proof of which is omitted) establishes property (i_V) and it can be shown that the next two lemmas suffice to establish properties (ii_V) and (iii_V).

Lemma 4.2 *If element $M(i,j)$ is assigned a color at line (9) of color_odd then $M(i,j) \in V(i_p)$.*

Lemma 4.3 *For $1 \leq j, j' \leq (n-1)/2 - i_p$, if $i_p + j + k * ((n-1)/2 - i_p) = i_p + j' + k' * ((n-1)/2 - i_p)$ then $j = j'$ and $k = k'$.*

Lemma 4.4 *For each $\hat{i}$, $i_p + 1 \leq \hat{i} \leq (n-1)/2 + i_p$, there are values j and k such that $i_p + j + k((n-1)/2 - i_p) = \hat{i}$ where $k \geq 0$ and $1 \leq j \leq (n-1)/2 - i_p$.*

Proof: Since $\hat{i} \geq i_p + 1 > i_p$ we can write $\hat{i} - i_p = q + t\left(\frac{n-1}{2} - i_p\right)$ for some integer t, $t \geq 0$ and some integer q, $0 \leq q < (n-1)/2 - i_p$. Suppose $q > 0$. Then take $j = q$ and $k = t$ and we are done. Suppose $q = 0$. If $t = 0$ then this implies that $\hat{i} = i_p$ thus contradicting the fact that $\hat{i} \geq i_p + 1 > i_p$. Thus $t \geq 1$ and so we write $\hat{i}$ as

$$\hat{i} = i_p + \left(\frac{n-1}{2}\right) + (t-1)\left(\frac{n-1}{2} - i_p\right)$$

and this shows that we can take $j = (n-1)/2$ and $k = t - 1 \geq 0$. ■

The above results allow us to conclude that properties (i_V), (ii_V) and (iii_V) are true. A similar series of lemmas (omitted) establishes properties (i_D), (ii_D) and (iii_D).

Thus each element of M is assigned a color exactly once. These lemmas together with Lemma 4.1 showing that the collection of routes assigned any given color form a tour, allows us to conclude that the procedure *color_odd* produces a valid monochromatic assignment of colors to all routes using $k_n = (n^2-1)/8$ colors.

4.2 Monochromatic Assignments for Even n

For connections of length less than $n/2$ the choice of shortest route is fixed and so we can discuss the problem of assigning colors to each such connection. Consider the procedure *color_even* that is exactly the same as the procedure *color_odd* in Fig. 1 except that the variable h is initially assigned the value $n/2-1$ rather than $(n-1)/2$. Then arguments analogous to those in Sect. 4.1 show that *color_even* monochromatically assigns colors to all connections of length less than $n/2$ and uses exactly $n(n-2)/8$ colors to achieve this assignment.

We then show that the connections of length $n/2$ can be routed so that it is possible to monochromatically assign colors to each such route using only $k_n - n(n-2)/8$ additional colors. Let C_{i,i^*} be the length $n/2$ connection from v_i to v_{i^*}. That is, $i^* = i \pmod{n/2}$. Then the n length $n/2$ connections can be partitioned into $n/2$ pairs $P_0, P_2, \ldots, P_{n/2-1}$ where $P_i = (C_{i,i^*}, C_{i^*,i})$. If i is even then route the two connections in P_i clockwise and otherwise route them counterclockwise. Then the two routes corresponding to the two connections of P_i form a tour and so the two routes can be assigned one color. It is shown that no more than $k_n - n(n-2)/8$ of the P_i's are routed clockwise and similarly no more than $k_n - n(n-2)/8$ of the P_i's are routed counterclockwise thus we conclude that the routes can be monochromatically assigned channels using no more than k_n colors.

References

1. A. Aggarwal, A. Bar-Noy, D. Coppersmith, R. Ramaswami, B. Schieber, and M. Sudan. Efficient routing and scheduling algorithms for optical networks. In *Proceedings of SODA*, pages 412–423, 1994.
2. S. B. Alexander, R. S. Banderant, D. Byrne, V. W. S. Chan, S. G. Finn, R. Gallager, B. S. Glance, H. A. Haus, P. Humblet, R. Jain I. P. Kaminow, M. Karol, R. S. Kennedy, A. Kirby, H. Q. Le, A. M. Saleh, B. A. Schofield, J. H. Shapiro, N. K. Shankaranarayanan, R. E. Thomas, R. C. Williamson, and R. W. Wilson. A precompetitive consortium on wide-band all-optical networks. *Journal of Lightwave Technology*, 11(5/6):714–735, May/June 1993.
3. M. C. Brain and P. Cochrane. Wavelength routed optical networks using coherent transmission. In *Proceedings of ICC*, volume 1, pages 26–31, 1988.
4. I. Chlamtac, A. Ganz, and G Karmi. Lightpath communications: An approach to high bandwidth optical wan's. *IEEE Transactions on Communications*, 40(7):1171–1182, July 1992.
5. M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W.H. Freeman and Co., San Francisco, CA, 1979.
6. P. E. Green. *Fiber-Optic Networks.* Prentice-Hall, Cambridge, MA, 1992.
7. G. Jeong and E. Ayanoglu. Comparison of wavelength-interchanging and wavelength-selective cross-connects in multiwavelength all-optical networks. Technical Report BLO113430-950927-29TM, AT&T Bell Laboratories, 1995.
8. K-C. Lee and V. O. K. Li. Routing and switching in a wavelength convertible optical network. In *Proceedings of IEEE INFOCOM*, volume 2, pages 578–585, 1993.
9. K. Sato, S. Okamoto, and H. Hadama. Network performance and integrity enhancement with optical path layer technologies. *IEEE Journal on Selected Areas of Communications*, 12(1):159–170, January 1994.
10. D. W. Smith and G. R. Hill. Optical processing in future coherent networks. In *Proceedings of GLOBECOM*, volume 2, pages 678–683, 1987.

This article was processed using the LaTeX macro package with LLNCS style