Efficient Routing and Wavelength Assignment for Reconfigurable WDM Ring Networks With Wavelength Converters

Li-Wei Chen, *Student Member, IEEE,* and Eytan Modiano, *Senior Member, IEEE*

***Abstract*—We consider the problem of wavelength assignment in reconfigurable WDM networks with wavelength converters. We show that for N-node P-port bidirectional rings, a minimum number of $\lceil PN/4 \rceil$ wavelengths are required to support all possible connected virtual topologies in a rearrangeably nonblocking fashion, and provide an algorithm that meets this bound using no more than $\lceil PN/2 \rceil$ wavelength converters. This improves over the tight lower bound of $\lceil PN/3 \rceil$ wavelengths required for such rings given in [1] if no wavelength conversion is available. We extend this to the general P-port case where each node i may have a different number of ports P_i, and show that no more than $\lceil \sum_i P_i/4 \rceil + 1$ wavelengths are required. We then provide a second algorithm that uses more wavelengths yet requires significantly fewer converters. We also develop a method that allows the wavelength converters to be arbitrarily located at any node in the ring. This gives significant flexibility in the design of the networks. For example, all $\lceil PN/2 \rceil$ converters can be collocated at a single hub node, or distributed evenly among the N nodes with $\min\{\lceil P/2 \rceil + 1, P\}$ converters at each node.**

***Index Terms*—Dynamic traffic, optical network, ring network, routing, wavelength assignment, wavelength division multiplexing (WDM).**

I. Introduction

In recent years, optical networks using wavelength division multiplexing (WDM) technology have emerged as an attractive solution for meeting rapidly growing demands for bandwidth. WDM allows the same fiber to carry many signals independently as long as each uses a different wavelength. Calls must therefore be routed and assigned to wavelengths such that no two calls use the same wavelength on the same link. This is known as the *routing and wavelength assignment* (RWA) problem. Calls are additionally subject to the *wavelength continuity constraint*, which requires that a call use the same wavelength on all hops unless wavelength conversion is available at intermediate nodes. If full conversion is available at all nodes, the WDM network is equivalent to a circuit-switched network; however, the high cost of wavelength converters often makes it desirable to keep the amount of conversion used in the network to a minimum.

Manuscript received Apr. 29, 2003; approved by IEEE/ACM Transactions on Networking Editor N. Shroff. This work was supported by the National Science Foundation (NSF) under Grant ANI-0073730.

The authors are with the Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: lwchen@mit.edu; modiano@mit.edu)

Digital Object Identifier 10.1109/TNET.2004.842214

There has been considerable work done in the area of finding efficient algorithms for the RWA problem. The literature adopts a number of different approaches to the problem. In the static traffic model, the traffic matrix representing the calls is fixed and does not change over time. In the dynamic traffic model, the traffic matrix is allowed to change over time to represent call arrivals and departures.

In the static model, the objective is typically to minimize the number of wavelengths, converters, or other cost parameters [2]. This problem was shown to be NP-complete in [3], and thus the literature has focused on the development of heuristics and bounds. Other approaches include attempting to maximize throughput for fixed capacity [4], to minimize congestion for a fixed traffic set [5], or to maximize the number of calls supported for a fixed number of wavelengths [6]. However, this approach is limited in that it does not allow dynamic call setup and removals.

The alternative is to use a dynamic model, where calls are allowed to arrive and depart over time. One method of modeling call dynamics is to adopt a statistical model for call arrival rates and holding times and design algorithms to minimize the call blocking probability. Numerous papers have focused on blocking probability analysis under various approximations for simple wavelength assignment algorithms such as the random algorithm [7]–[12] and first-fit [13]. However, due to the large state-space size of the problem, the blocking probability of a WDM network for more sophisticated algorithms is extremely difficult to analyze. As a result, most statistical algorithms rely on simplifying approximations and heuristics [14].

An alternative approach considers designing the network to accommodate any traffic matrix from an admissible set. Call arrivals or departures are equivalent to transitioning from one traffic matrix to another. If the transitions can be accommodated without rearranging any calls, the RWA algorithm is called *wide-sense nonblocking*; algorithms which require call rearrangement are called *rearrangeably nonblocking*. For example, [15] considers a traffic set such that the maximum load on each link is bounded by some constant, and attempts to minimize the number of wavelengths used at that given load; [16] works on minimizing the wavelength converter usage for networks using a number of wavelengths equal to the maximum link load. Another approach is taken in [1] by admitting any traffic matrix where each node uses at most P ports. It is shown that for the case of a bidirectional ring with N nodes and P ports, a lower bound of $\lceil PN/3 \rceil$ wavelengths is required to support the worst-case traffic set if no wavelength conversion is employed. Moreover, in [1] a rearrangeably nonblocking RWA algorithm is provided which achieved this bound. An online version based on these ideas was presented in [17] which additionally attempts to minimize the number of rearrangements required; this algorithm was later extended from rings to torus networks in [18]. The P-port model is very practical since the admissible set is based on actual device limitations in the network. In this paper, we investigate new rearrangeably nonblocking RWA schemes for this admissible set where wavelength conversion is available.

A. System Model

We consider a bidirectional ring with N nodes. Adjacent nodes are connected by two fibers: one supporting wavelengths travelling in the clockwise direction, the other supporting wavelengths in the counterclockwise direction. The two fibers are represented by a single bidirectional link, where each link can support calls travelling in both directions on every wavelength.

A wavelength converter, if available at a given node, can be used to switch a call arriving to that node on one wavelength onto a different wavelength departing the node. If no conversion is employed, a call passing through a node on one wavelength must exit the node on the same wavelength. The cost of providing wavelength conversion from one wavelength to another is assumed to be fixed and independent of the frequency separation between the wavelengths. A traffic matrix or traffic set consists of a set of calls that need to be set up in the network. Each call consists of a source and destination pair. A traffic set is *connected* if the directed graph corresponding to the set of source-destination pairs is connected. In a *single-port* network, each node is considered to have a single tunable optical transmitter and receiver. Hence each node may at most originate one call (using any available wavelength) and receive one call (on any wavelength, possibly different from the one used by the transmitter). In a P-*port* network, each node i has P_i transmitters and receivers, and hence can transmit and receive P_i different calls. P-port networks can be either *symmetric*, where $P_i = P$ for all nodes, or *asymmetric*, where P_i can differ for each node. This is a natural problem to consider since equipment constraints limit the number of ports each node has available. The set of all traffic matrices which satisfy the P-port requirement is called the *admissible set*. Routing and assigning wavelengths to each of these traffic matrices is the RWA problem, considered in this paper.

We consider the problem of supporting any admissible traffic set in a P-port network in a rearrangeably nonblocking fashion. In this context, there are a number of metrics which are relevant to evaluating the performance of a RWA algorithm. One is the worst-case number of wavelengths required by the algorithm – the smaller the number, the better. Another is the total number of wavelength converters the algorithm uses. Since converters are expensive, an algorithm that uses converters sparingly is preferred. Finally, in general the converter requirements may be different at each node. Certain distributions may be more desirable than others depending on the design criteria: for example, in some cases, we may want a *hub* design where all converters are placed at a single node; in others we may prefer the converters to be distributed equally at all nodes. We consider algorithms which attempt to design a RWA for these metrics.

In Section II, we derive a lower bound on the number of wavelengths required to support the worst-case traffic set, and present two RWA schemes for both connected and unconnected traffic sets in single-port networks: an optimal algorithm which uses the minimum possible number of wavelengths to support all traffic sets, and a suboptimal algorithm which uses more wavelengths but requires significantly fewer converters. These results are extended to multi-port networks in Section III. In Section IV we develop a method for changing the location of wavelength converters in a given RWA, and apply the method to the algorithms in the previous sections.

II. Single-Port Ring Networks

A. The $\lceil N/4 \rceil$ Algorithm for Connected Rings

We consider here the case of a single-port network, and require that the RWA algorithm be able to route any connected traffic set in a rearrangeably nonblocking fashion. Our initial goal is to design a RWA algorithm which minimizes the number of wavelengths used. The following theorem gives a lower bound on the number of wavelengths required by the worst-case traffic set for this network.

Theorem 1: For a single-port N-node bidirectional ring, at least $\lceil N/4 \rceil$ wavelengths are required by the worst-case traffic set for N even, and $\lceil (N-1)/4 \rceil$ wavelengths for N odd.

Proof: Consider the case where N is even, and envision a cut which divides the network into two sets of $N/2$ nodes each. Since the nodes were formed in a ring, this cut consists of two links. Consider a traffic set where each of the $N/2$ nodes in one set wishes to communicate to one of the nodes in the other set. In this case, a total of $N/2$ calls must cross the cut in either direction, for a total of N calls. Since each link in the cut can support at most two calls on a single wavelength (one clockwise, one counterclockwise), a minimum of $\lceil N/4 \rceil$ wavelengths are required to support the calls across the cut. Similar reasoning for N odd gives a bound of $\lceil (N-1)/4 \rceil$. ∎

It is worth noting that this bound cannot be achieved by a simple routing scheme such as shortest-path. To see this, consider a ring with an even number of nodes N, and number the nodes in increasing order from 1 to N in the clockwise direction. Consider the traffic set where each node n_i sends a call to node $n_{i \oplus [(N/2)-1]}$. (We use $\oplus$ to denote addition modulo N.) Then shortest-path would route all calls in the clockwise direction, with each call requiring $(N/2)-1$ hops to accommodate it. Since there are N calls total, this would require at least $N \cdot (N/2-1)/N = (N/2) - 1$ wavelengths to support it.

We next describe the operation of our first RWA algorithm and assert that it is optimal in the sense that it requires no more than the lower bound of $\lceil N/4 \rceil$ wavelengths. The proof follows the description.

Consider an arbitrary connected traffic set $\{c_1, c_2, \ldots, c_N\}$ consisting of source-destination pairs c_i. We term a pair of calls *adjacent* if the destination node of the first call is the source node of the second. In a connected traffic set, it is always possible to traverse all calls in the traffic set in adjacent order; i.e., there are no sub-cycles within the traffic set. Therefore without loss of generality we can renumber the calls so that they are indexed in adjacent order; that is, c_i is adjacent to $c_{i\oplus 1}$ for every i.

Denote the number of hops required to route a particular call c_i in the clockwise direction by L_i. Denote the average number of hops required in the clockwise direction by

$$\bar{L} = \frac{\sum_{i=1}^{N} L_i}{N}.$$

Then the algorithm is as follows:

THE $\lceil N/4 \rceil$ ALGORITHM

1) TRAFFIC SET PARTITIONING: Let $k = \min\{\lfloor N^2/4\bar{L}\rfloor, N\}$. Find a set of k adjacent calls with average clockwise hop length $\tilde{L}$ less than or equal to $\bar{L}$. Call this set the *clockwise set*. Designate all calls not contained in the clockwise set to be members of the *counterclockwise set*. (We will shortly show that such sets always exist.)
2) ROUTING: Route all calls in the clockwise set in the clockwise direction. Route all calls in the counterclockwise set in the counterclockwise direction.
3) WAVELENGTH ASSIGNMENT (CLOCKWISE SET): Assign wavelengths to calls using a *forward pass* and a *reverse pass* as follows: Index all calls c_m in the clockwise set in adjacent order. Index the wavelengths λ_n in arbitrary order. Initialize $i = 1$ and $j = 1$.
   a) FORWARD PASS: In this part, beginning with the first call and proceeding in adjacent order, assign as many calls as possible to the first wavelength without using conversion. When a call cannot be fully assigned to the wavelength, assign it entirely to the next wavelength (without conversion) and repeat, until all $\lceil N/4 \rceil$ wavelengths are used. This is made explicit below:
      i) Assign call c_i entirely to λ_j without using any conversion.
      ii) Increment i: $i \leftarrow i + 1$.
      iii) If call c_i can be assigned entirely to λ_j without conversion, goto (i). Otherwise continue.
      iv) Increment j: $j \leftarrow j + 1$.
      v) If $j \leq \lceil N/4 \rceil$, goto (i). Otherwise stop.
   b) REVERSE PASS: In this part, the remaining calls are assigned to the wavelengths in the reverse of the order they were filled in the forward pass, using converters as necessary. More explicitly:
      i) Assign as much of the unassigned portion of call c_i to λ_j as possible.
      ii) If c_i is completely assigned, increment i and goto (i). Otherwise continue.
      iii) Using a wavelength converter, convert the last hop of c_i allocated in (i) from λ_j to λ_{j-1}.
      iv) Decrement j: $j \leftarrow j - 1$.
      v) If all calls have been assigned, stop. Otherwise goto (i).
4) WAVELENGTH ASSIGNMENT (COUNTERCLOCKWISE SET): Repeat Step 3 with the counterclockwise set in the counterclockwise direction.

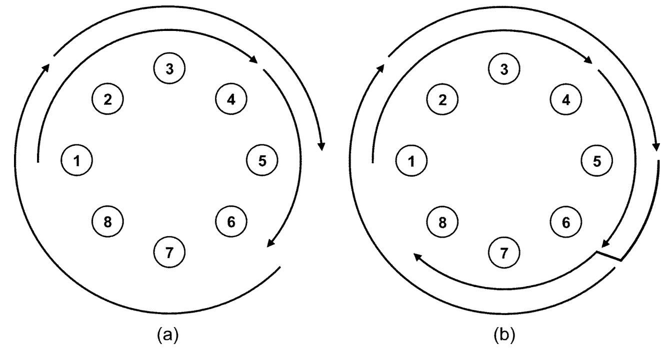


Fig. 1. (a) The routing and wavelength assignment of calls in the clockwise set after the forward pass. The inner arrows represent calls on λ_1, the outer arrows are calls on λ_2. (b) The complete RWA on the clockwise direction after the backward pass.

We will refer to this as the $\lceil N/4 \rceil$ algorithm. The following example illustrates the use of the $\lceil N/4 \rceil$ algorithm for a particular traffic set.

Example 1: Consider an 8-node ring, where $\lceil N/4 \rceil = 2$. Number the nodes from 1 to 8 in the clockwise direction. Consider a traffic set consisting of the following calls, listed in adjacent order: (1,4), (4,6), (6,2), (2,5), (5,8), (8,3), (3,7), and (7,1). We will apply the $\lceil N/4 \rceil$ algorithm to this problem.

The average clockwise hop length $\bar{L} = 3$, and $k = \min\{\lfloor N^2/4\bar{L}\rfloor, N\} = \min\{\lfloor 16/3\rfloor, 8\} = \min\{5, 8\} = 5$. Choose the clockwise set to be the set of calls {(1,4), (4,6), (6,2), (2,5), (5,8)}, with average hop length $\tilde{L} = (3+2+4+3+3)/5 = 3 \leq \bar{L}$. The counterclockwise set then consists of the remaining calls, {(8,3), (3,7), (7,1)}. Note that the average hop length obeys $\hat{L} = (3+4+2)/3 \geq \bar{L}$ in the clockwise direction.

In the forward pass on the clockwise set, calls (1,4) and (4,6) are assigned to the first wavelength, while (6,2) and (2,5) are assigned to the second wavelength. This situation is shown in Fig. 1(a). In the reverse pass, the final call (5,8) is assigned partly on each wavelength and employs a converter at node 6. The final RWA for the clockwise set is shown in Fig. 1(b).

In the forward pass on the counterclockwise set, calls (8,3) and (3,7) are assigned to the first and second wavelengths, respectively. In the reverse pass, (7,1) is assigned partly to both and again requires a converter.

We make two claims regarding this algorithm. First, it is always possible to find a set of $k = \min\{\lfloor N^2/4\bar{L}\rfloor, N\}$ adjacent calls with average clockwise hop length less than or equal to $\bar{L}$. Second, using this algorithm, any admissible traffic set requires at most $\lceil N/4 \rceil$ wavelengths and $\lceil N/2 \rceil - 2$ converters. These claims will be formalized as Lemma 1 and Theorem 2.

Lemma 1: There exists a set of n adjacent calls with average clockwise hop length $\tilde{L}$ less than or equal to the average clockwise hop length of the entire traffic set $\bar{L}$, for any $0 \leq n \leq N$. Furthermore, the $N - n$ calls in the complement of that set have average clockwise hop length $\hat{L} \geq \bar{L}$.

Proof: We will conduct a proof by contradiction. Suppose there did not exist any set of n adjacent pairs with average hop length less than $\bar{L}$. In particular, this would imply that

$$\frac{1}{n} \cdot (L_1 + \cdots + L_n) > \bar{L}$$

$$\frac{1}{n}\cdot(L_2+\cdots+L_{n+1})>\bar{L}$$
$$\cdots$$
$$\frac{1}{n}\cdot(L_{N-n+2}+\cdots+L_N+L_1)>\bar{L}$$
$$\cdots$$
$$\frac{1}{n}\cdot(L_N+L_1+\cdots+L_{n-1})>\bar{L}.$$

Summing the entire set of N inequalities, we obtain

$$L_1+\cdots+L_N>\bar{L}N$$

where the coefficient of each term L_i is unity, since each L_i is involved in exactly n of the inequalities and is scaled by a factor of $1/n$. Equivalently,

$$\frac{1}{N}\cdot(L_1+\cdots+L_N)>\bar{L}.$$

But since by definition $\bar{L}$ is the average hop length, this cannot be true. Hence there must exist a set of n adjacent pairs with average hop length less than $\bar{L}$.

The second half of the proof also uses contradiction. Suppose for the remaining $N-n$ calls, the average clockwise hop length $\hat{L}<\bar{L}$. From the definitions of $\hat{L}$ and $\tilde{L}$, we have that

$$\hat{L}=L_{n+1}+\cdots+L_N<(N-n)\bar{L}$$
$$\tilde{L}=L_1+\cdots+L_n\le n\bar{L}.$$

Combining the preceding two inequalities and dividing by N, we then obtain

$$\frac{1}{N}\cdot(L_1+\cdots+L_N)<\bar{L}$$

which contradicts the definition of $\bar{L}$ being the average hop length. ■

For our purposes, we will mainly be interested in applying Lemma 1 for the case of $n=k$ in the proof of the following theorem.

Theorem 2: Given any connected traffic set, the $\lceil N/4\rceil$ algorithm requires only $\lceil N/4\rceil$ wavelengths and at most $\lceil N/2\rceil-2$ converters.

Proof: By Lemma 1, it is always possible for the algorithm to find valid clockwise and counterclockwise sets. Consider first the clockwise set. For simplicity, consider those cases where the total number of wavelengths $N/4$ is an integer. (For all other cases, fictitious nodes can be added to increase $N/4$ to the nearest integer.) First note that $N/4$ wavelengths in an N-hop ring can support $N^2/4$ contiguous hops of traffic. By choice of the clockwise set, the average clockwise hop length in the clockwise direction $\tilde{L}\le\bar{L}$. Then the total number of hops required to accommodate the clockwise set, denoted by D_C, is

$$\begin{aligned}D_C&=k\tilde{L}\\&\le k\bar{L}\\&\le\left\lfloor\frac{N^2}{4\bar{L}}\right\rfloor\cdot\bar{L}\\&\le\frac{N^2}{4}.\end{aligned}$$

Since all required hops are contiguous due to the adjacency of all calls in the set, the clockwise set fits in $N/4$ wavelengths.

Next consider the counterclockwise set, which contains the remaining $N-k$ calls. If $k=N$, then $N-k=0$ and the counterclockwise set is empty and requires no wavelengths, completing the proof. Therefore assume $k=\lfloor N^2/4L\rfloor$. Denote the average clockwise hop length $\hat{L}$; this implies that the average counterclockwise hop length is $N-\hat{L}$. Since by Lemma 1 $\hat{L}\ge\bar{L}$, it must be that the average counterclockwise hop length $N-\hat{L}\le N-\bar{L}$. Denote the total number of contiguous hops required to accommodate the counterclockwise set by D_W. Then

$$\begin{aligned}D_W&=(N-k)\cdot(N-\hat{L})\\&\le(N-k)\cdot(N-\bar{L})\\&=\left(N-\left\lfloor\frac{N^2}{4\bar{L}}\right\rfloor\right)\cdot(N-\bar{L}).\end{aligned}$$

We show in Appendix A that for N even, the last quantity is maximized at $\bar{L}=N/2$, giving us

$$D_W\le\left(N-\left\lfloor\frac{N^2}{4\bar{L}}\right\rfloor\right)\cdot(N-\bar{L})\le\frac{N^2}{4}$$

which also fits in the $\lceil N/4\rceil$ wavelengths. Note that there is no loss of generality in the assumption of N even, as explained earlier and in the Appendix .

By construction, the $\lceil N/4\rceil$ algorithm requires up to one converter on each wavelength (except the last) in each direction, for a total of $2\lceil N/4\rceil-2$ converters. Additionally, consider the location of the converters: each converter, where needed, is located at the destination node of the last call on each wavelength after the forward pass on the clockwise and counterclockwise sets. Since we are dealing with a single-port network, each node is the destination of no more than a single call. This implies that no node requires more than a single converter at most. ■

Later, in Section IV, we will show how the wavelength assignment can be modified to distribute the $2\lceil N/4\rceil-2$ converters almost arbitrarily among all nodes in the ring.

B. The $2\lceil N/7\rceil$ Algorithm for Connected Rings

Although the $\lceil N/4\rceil$ algorithm achieves the minimum number of wavelengths, it may require as many as $2\lceil N/4\rceil-2$ converters to do so. Since converters may be costly, it is desirable to reduce the number of converters required. In [1] an algorithm is provided that does not require converters but uses $\lceil N/3\rceil$ wavelengths. Motivated by a desire to find a compromise between these two extremes, we present our next algorithm that requires $2\lceil N/7\rceil$ wavelengths and only $\lceil N/7\rceil$ converters.

We will begin by restating a result from [1] regarding the routing of adjacent pairs and giving a new lemma on routing adjacent triplets. Then, using these results, we will give an algorithm which divides the connected traffic set into smaller sets of 7 adjacent calls and routes each set of 7 calls onto two wavelengths (in each direction).

Lemma 2: Given an adjacent pair of calls, it is possible to fit the calls onto a single wavelength in either the clockwise or counterclockwise direction with no wavelength conversion.

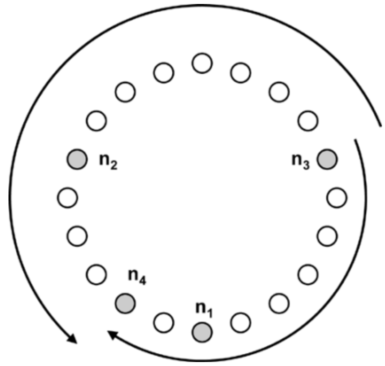


Fig. 2. Beginning at node n_3, since we first encounter node n_1 before n_4 when travelling in the clockwise direction, we must encounter n_4 before n_1 when travelling in the counterclockwise direction.

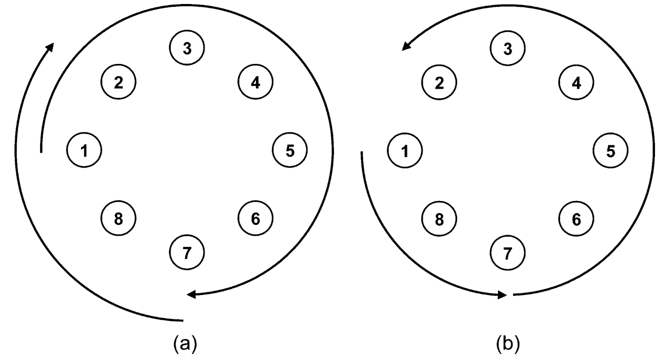


Fig. 3. (a) This adjacent pair cannot be placed on a single wavelength in the clockwise direction. (b) Therefore by Lemma 2, it can fit without converters on a single wavelength in the counterclockwise direction.

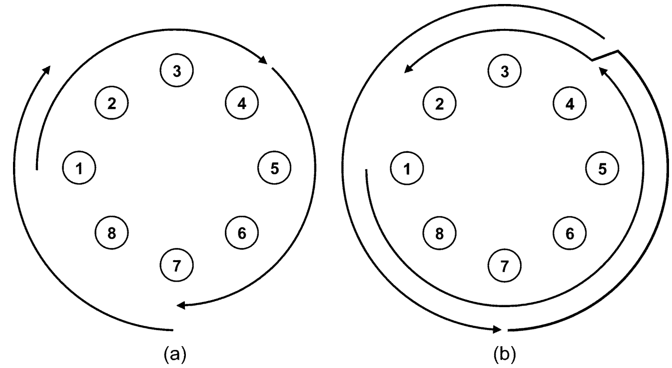


Fig. 4. (a) The adjacent triplet (n_1, n_4), (n_4, n_7), (n_7, n_2) cannot be placed on a single wavelength in the clockwise direction. (b) Therefore by Lemma 3, it can fit on two wavelengths in the counterclockwise direction using only a single converter. The converter is required at node 4 in this case. Notice also that the triplet can fit using two wavelengths in the clockwise direction.

Proof: See [1]. ■

Lemma 3: Given a direction around the ring and given an adjacent triplet of calls, if it is not possible to fit the calls into a single wavelength (using no converters) in that direction, then it is possible to fit the calls into two wavelengths (using a single converter) in the opposite direction.

Proof: Denote the calls by their source-destination pairs as follows: (n_1, n_2), (n_2, n_3), and (n_3, n_4). Without loss of generality, suppose by Lemma 2 that (n_1, n_2) and (n_2, n_3) fit on a single wavelength in the clockwise direction. (If the opposite is true, then simply reverse the clockwise/counterclockwise directions to follow.) We prove the lemma first for the choice of the clockwise direction, then the counterclockwise.

CLOCKWISE: Suppose the choice of direction was clockwise. If all three calls can be routed in the clockwise direction, then this part of the proof is complete. Suppose they cannot; i.e., part of the path (n_3, n_4) overlaps part of the path (n_1, n_2) in the clockwise direction. This implies that, travelling in a clockwise direction from node n_3, we first encounter node n_1 before node n_4. Reversing the directions, it must therefore also be the case that travelling in a counterclockwise direction from n_3, we first encounter node n_4 before node n_1. This is illustrated in Fig. 2.

We can route (n_1, n_2) and (n_2, n_3) each onto separate wavelengths λ_1 and λ_2 in the counterclockwise direction. This leaves the links between n_2 to n_1 on λ_1 and n_3 to n_2 free on λ_2. Since travelling in the counterclockwise direction we reach node n_4 before n_1, the third call (n_3, n_4) can fit into the free links on λ_1 and λ_2 in the counterclockwise direction using a converter at node n_2.

COUNTERCLOCKWISE: Next consider if the choice was counterclockwise. It is not possible to fit all calls into a single wavelength in this direction, so therefore we must show it is possible to fit all calls in two wavelengths in the clockwise direction. This is done by noting that since by assumption the first two calls can fit on a single wavelength in the clockwise direction, the third can fit alone on a second wavelength. ■

Figs. 3 and 4 illustrate examples of applying Lemmas 2 and 3, respectively. We will now use the two preceding lemmas to describe a method for fitting any set of 7 adjacent calls onto at most two wavelengths.

Theorem 3: Given a set of 7 adjacent calls, the entire set can be routed using at most two wavelengths (in each direction).

Proof: We will provide a proof by construction. Consider the first four adjacent calls. Divide them into two adjacent pairs. By Lemma 2, each pair can be routed using a single wavelength in either the clockwise or counterclockwise direction. First suppose that the two wavelengths are in different directions. Then they can share the same wavelength, and the first four paths can be routed using a single wavelength. Of the remaining three calls, by Lemma 2 the first adjacent pair can again be fit on a single wavelength in one direction; placing the remaining call on the same wavelength in the opposite direction completes the construction in this case.

Next suppose that the first two pairs can only fit on single wavelengths in the same direction. Without loss of generality, let this direction be clockwise. Consider the remaining adjacent triplet.

If these calls can be placed onto a single wavelength in the clockwise direction, then do so. Also place the first pair on a second wavelength in the clockwise direction. Then place the two calls in the second pair on the same two wavelengths in the counterclockwise direction, each using their own wavelength.

If the last three calls cannot be placed onto a single wavelength in the clockwise direction, then by Lemma 3 they can be placed onto at most two wavelengths in the counterclockwise direction. The first two pairs can then be routed onto the same two wavelengths in the clockwise direction, each pair using its own wavelength. ■

In general, we can route any connected traffic set by dividing it into adjacent sets of 7 calls and applying the construction in the proof of Theorem 3 to each set. We will call this the $2\lceil N/7\rceil$ algorithm.

THE $2\lceil N/7\rceil$ ALGORITHM

1) Divide the traffic set into $c = \lceil N/7\rceil$ adjacent sets of 7, each denoted by C_j, $1 \le j \le c$. Let $i = 1$.
2) Route each set of 7 calls using 2 wavelengths, following the proof of Theorem 3, for a total of $2\lceil N/7\rceil$ wavelengths.

Converter Requirements: During the RWA construction, the traffic set is divided into $\lceil N/7\rceil$ sets of 7 adjacent calls; each set of 7 calls uses at most a single converter. Using these facts, we can show that the total number of converters required is upper-bounded by $\lfloor N/7\rfloor$.

To see why we can use only $\lfloor N/7\rfloor$ rather than $\lceil N/7\rceil$ converters, we need to consider two cases: where N is and is not divisible by 7. Supposing N is divisible by 7, $\lfloor N/7\rfloor = \lceil N/7\rceil$ and the distinction is irrelevant. Next suppose N is not divisible by 7. Then the first $\lceil N/7\rceil - 1$ sets require at most $\lceil N/7\rceil - 1 = \lfloor N/7\rfloor$ converters. The last set has at most 6 adjacent calls. (If it has less, insert fictitious calls.) Further divide this set into two sets of 3 adjacent calls. Each set of 3 calls can be routed using a single wavelength without conversion by putting the first two adjacent calls onto a single wavelength in one direction without conversion (guaranteed by Lemma 2) and putting the remaining call in the other direction on the same wavelength.

The converter in each set, if required, is located at the destination of one of the calls. Since we are considering a single-port network wherein each node form the destination of only one call in the traffic set, no node requires more than one converter. We later show in Section IV how the wavelength assignment can be modified to distribute the $\lfloor N/7\rfloor$ converters almost arbitrarily among all nodes.

C. Handling Unconnected Traffic Sets

Thus far we have limited our discussion to connected traffic sets. We next consider unconnected traffic sets; that is, traffic sets where in the corresponding directed graph there exist nodes which do not communicate. For single-port traffic, we will see that this implies that the traffic set is composed of a number of cycles.

We consider only *maximal* traffic sets; i.e., traffic sets containing the maximum number of calls given the single-port restriction. Note that any nonmaximal traffic set can be converted to a maximal set by adding fictitious calls; hence it is sufficient to consider the RWA of maximal sets. We can construct the cycles as follows:

1) Initialize $i = 1$.
2) Choose an arbitrary node, called the cycle start node. Find the call originating at that node. Move to the destination of that call. Now find the call originating at this new node, and move to the destination of that call. Repeat. By the maximal assumption, each node must originate a call, so this is always possible. The cycle is complete when the start node is revisited. Designate all calls traversed in this step as members of the cycle C_i.
3) Remove all calls in C_i from the traffic set. By the single-port assumption, since each node encountered in the previous step is the source and destination of some call in C_i, they are not involved in any remaining calls in the traffic set.
4) If the traffic set is not yet empty, increment $i \leftarrow i + 1$ and goto Step 2.

This construction divides the traffic set into cycles involving disjoint sets of nodes. Next we will give a method for dealing with traffic sets with cycles by using an additional wavelength to turn it into a different RWA problem for a connected traffic set that does not contain cycles. The connected traffic set can then be processed using either of the previous algorithms.

Theorem 4: Suppose there exists an algorithm that uses at most W wavelengths for any admissible connected traffic set in a single-port ring network. Then any admissible traffic set with cycles can be routed using at most $W + 1$ wavelengths with the addition of a number of converters equal to the number of cycles.

Proof: The proof is by construction using the following algorithm.

Step 1 – CYCLE FORMATION: Consider a traffic set with c cycles. Group the calls into sets based on which cycle they belong to. Number these cycles $C_1, C_2, \ldots, C_c$. From each set, arbitrarily choose a single call and denote the source and destination nodes of that call by s_i and d_i, respectively, for the set i. Without loss of generality, renumber the cycles so that $d_1, \ldots, d_c$ are in *counterclockwise order*; i.e., after renumbering, travelling counterclockwise around the ring beginning with d_1, one encounters each d_i in order of increasing index i.

Step 2 – SUPERCYCLE FORMATION: The idea is that we will break each cycle at the call (s_i, d_i) and connect it to the next cycle, thus forming a single connected *supercycle*. Consider a given cycle C_i. Remove the call (s_i, d_i) from the traffic set, and replace it with a new call $(s_i, d_{i\oplus 1})$. This connects all nodes in cycle C_i with cycle $C_{i\oplus 1}$. Repeat for all cycles. At the end of this procedure, we have formed a new traffic set called the supercycle, denoted by C_S. Note that the supercycle is also a maximal, admissible traffic set that obeys the single-port restrictions, since essentially all it did was permute the destinations of the various (s_i, d_i) calls of the original set.

Step 3 – RESIDUAL SET: We next need to add a set of additional calls, which we call the *residual set* C_R, to make $C_S \cup C_R$ equivalent to the original traffic set. The residual set consists of calls $(d_{i\oplus 1}, d_i)$ for $1 \le i \le c$. Then for a given cycle C_i, we can combine the calls $(s_i, d_{i\oplus 1})$ and $(d_{i\oplus 1}, d_i)$ from C_S and C_R, respectively, to form the original call (s_i, d_i). At most a single converter is needed at $d_{i\oplus 1}$ if the two calls are on different wavelengths.

Step 4 – RWA OF C_S AND C_R: The RWA algorithm for connected traffic sets can be used on C_S using at most W wavelengths by assumption. Thus it remains only to show that C_R can be fit onto a single additional wavelength. The calls in this set consist of $(d_c, d_{c-1}), (d_{c-1}, d_{c-2}), \ldots, (d_3, d_2), (d_2, d_1)$. Note that this traffic set simply traverses all the d_i's in descending order. Since the d_i's were chosen in counterclockwise order by ascending i, it follows that they must be in clockwise order by

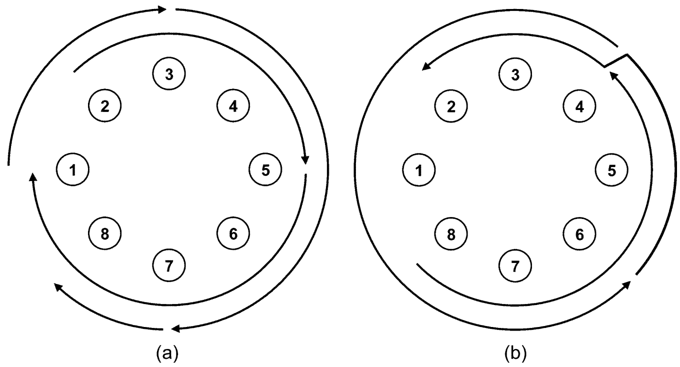


Fig. 5. The RWA for superset T_S of Example 2.

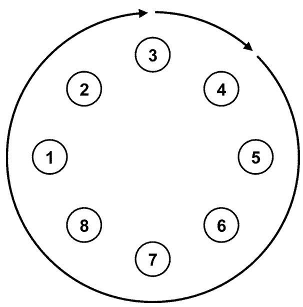


Fig. 6. The RWA of residual set T_R.

descending i. Therefore all calls in T_1 can be fit onto a single wavelength in the clockwise direction. ■

Corollary 1: The $\lceil N/4 \rceil$ algorithm can handle unconnected traffic sets using at most $\lceil N/4 \rceil + 1$ wavelengths.

Corollary 2: The $2\lceil N/7 \rceil$ algorithm can handle unconnected traffic sets using at most $\lceil 2N/7 \rceil + 1$ wavelengths.

The following example demonstrates the application of this approach to a traffic set with two cycles.

Example 2: Consider an 8-node ring with nodes numbered from 1 to 8 in the clockwise direction. Consider a traffic set consisting of the following calls, listed in adjacent order: (1,4), (4,6), (6,2), (2,5), (5,1), (8,3), (3,7), and (7,8). Note that the traffic set has two cycles: $C_1 = \{(1,4),(4,6),(6,2),(2,5),(5,1)\}$, and $C_2 = \{(8,3),(3,7),(7,8)\}$. We arbitrarily choose the calls (1,4) and (8,3) from C_1 and C_2, respectively. Then $d_1 = 4$, and $d_2 = 3$. Since there are only two nodes, they are trivially in counterclockwise order and we do not need to renumber the cycles.

In addition to the previously noted values of d_1 and d_2, we also have that $s_1 = 1$ and $s_2 = 8$. Following the preceding approach, in the superset call (1,4) becomes $(s_1, d_2) = (1,3)$. Similarly, (8,3) becomes $(s_2, d_1) = (8,4)$. The superset is $T_S = \{(1,3),(4,6),(6,2),(2,5),(5,1),(8,4),(3,7),(7,8)\}$. Reordered into adjacent order, we have $T_S = \{(8,4),(4,6),(6,2),(2,5),(5,1),(1,3),(3,7),(7,8)\}$.

The residual set is $T_R = \{(d_1,d_2),(d_2,d_1)\} = \{(3,4),(4,3)\}$.

We can now route T_S using any algorithm we choose. Here we will route it using the $\lceil N/4 \rceil$ algorithm. The set T_R can by choice fit into a single wavelength. The RWA for T_S and T_R are illustrated in Figs. 5 and 6 respectively. Finally, the calls that

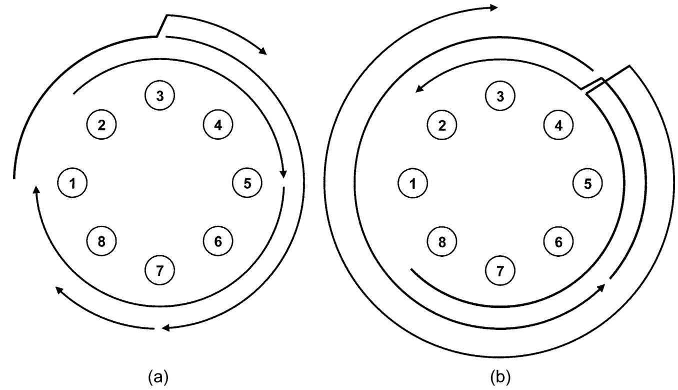


Fig. 7. (a) and (b) show the final RWA for Example 2 in the clockwise and counterclockwise directions, respectively. Note that although the call (8,3) in (b) ended up being routed partly in the counterclockwise direction and partly in the clockwise direction, the hops in the clockwise direction do not require an additional wavelength since those hops are free on one of the existing wavelengths in (a). Also note that the RWA could be simplified by routing call (8,3) entirely in the clockwise direction, although this does not result in a savings in total wavelengths used.

were split during the creation of T_S and T_R are reconnected using wavelength converters in Fig. 7.

Converter Requirements: By construction, one converter is required per cycle in addition to any converter requirements by the RWA algorithm.

III. Multi-Port Ring Networks

A. The $\lceil PN/4 \rceil$ Algorithm

1) Symmetric Multi-Port Networks: We first consider the case of connected symmetric P-port networks. By symmetric, we mean that each node has the same number of ports P. In such a network, each node has P transmitters and receivers, and can therefore send and receive P calls. Since each node is the source of at most P calls, and there are N nodes, a full traffic set contains at most PN calls. Again using a cut-set bound, it is apparent that a minimum of $\lceil PN/4 \rceil$ wavelengths are required to support the worst-case traffic set.

If the logical topology is connected, then the directed graph contains a directed Euler trail [19] which contains all edges of the graph. By finding and following the Euler trail, we can obtain the PN calls in adjacent order. We can apply a modified version of the $\lceil N/4 \rceil$ algorithm, which we will call the $\lceil PN/4 \rceil$ algorithm, to this traffic set.

THE $\lceil PN/4 \rceil$ ALGORITHM

1) TRAFFIC SET PARTITIONING: Let $k = \min\{\lfloor PN^2/4\bar{L} \rfloor, PN\}$. Find a set of k adjacent calls with average clockwise hop length $\tilde{L}$ less than or equal to $\bar{L}$. Call this set the *clockwise set*. Designate all calls not contained in the clockwise set to be members of the *counterclockwise set*.
2) ROUTING: Route all calls in the clockwise set in the clockwise direction. Route all calls in the counterclockwise set in the counterclockwise direction.
3) WAVELENGTH ASSIGNMENT: Assign wavelengths to calls using a forward and reverse pass on both the clockwise and counterclockwise sets, as in the original $\lceil N/4 \rceil$ algorithm.

This algorithm requires at most $\lceil PN/4 \rceil$ wavelengths. The proof follows the same procedure as Section II-A.

For the $\lceil PN/4 \rceil$ algorithm, up to one converter on each wavelength (except the last) is required in each direction, for a total of $2\lceil PN/4 \rceil - 2$ converters. However, since we have a P-port network, similar examination of the construction of the wavelength assignment shows that since each node can be the destination of up to P calls, it may require at most P converters. Again, in Section IV we will show how the wavelength assignment can be modified to distribute the $2\lceil PN/4 \rceil - 2$ converters nearly arbitrarily among all nodes. In particular, a modified wavelength assignment can be given that requires no more than $\min\{\lceil P/2 \rceil + 1, P\}$ converters per node.

The $\lceil PN/4 \rceil$ algorithm can also be applied to general unconnected networks containing cycles by using the approach of Section II-C, where one additional wavelength is used to convert the traffic set into a connected traffic set.

2) General Multi-Port Networks: We next consider general networks where each node i has P_i ports, and is able to transmit and receive at most P_i calls. Under this model, the nodes can now be heterogeneous, and consequently it allows the model a great deal of generality.

Let $P_{tot} = \sum_{i=1}^{N} P_i$ be the total number of calls in the system. The following theorem states that for any admissible traffic set, connected or unconnected, it is possible to obtain a RWA for any admissible traffic set using at most $\lceil P_{tot}/4 \rceil + 1$ wavelengths.

Theorem 5: For a general multi-port network with a traffic set containing a maximum of P_{tot} calls, the $\lceil PN/4 \rceil$ algorithm requires at most $\lceil P_{tot}/4 \rceil + 1$ wavelengths to provide a RWA for any arbitrary admissible traffic set.

Proof: First, if the traffic set is unconnected, we use an approach similar to the one in Section II-C to turn it into a connected set. This requires using a single additional wavelength in the clockwise direction.

From this point on, we can assume that the traffic set is connected, and apply the $\lceil PN/4 \rceil$ algorithm, with the only difference being that the clockwise set is chosen to be of size $k = \min\{\lfloor P_{tot}N/4\bar{L} \rfloor, P_{tot}\}$ calls. By a proof similar to the one used for Lemma 1, it can be shown that the existence of a clockwise and counterclockwise set is guaranteed. Thus it remains only to show that no more than $\lceil P_{tot}/4 \rceil + 1$ wavelengths are required by both the clockwise and counterclockwise sets.

First consider the clockwise set. Since the total number of calls is P_{tot}, and the average (clockwise) hop length is at most $\bar{L}$, then the number of contiguous clockwise hops required is

$$\begin{aligned} D_C &\leq \left\lfloor \frac{P_{tot}N}{4\bar{L}} \right\rfloor \cdot \bar{L} \\ &\leq \left(\frac{P_{tot}N}{4\bar{L}} \right) \cdot \bar{L} \\ &= \frac{P_{tot}N}{4}. \end{aligned}$$

Since each wavelength can support N contiguous hops of traffic, no more than $\lceil (P_{tot}N/4)/N \rceil = \lceil P_{tot}/4 \rceil$ wavelengths are required in the clockwise direction.

Next consider the counterclockwise direction. Again if $k = P_{tot}$ the counterclockwise set is empty, so the only case of interest is when $k = \lfloor P_{tot}N/4\bar{L} \rfloor$. Here the total number of calls is $P_{tot} - \lfloor P_{tot}N/4\bar{L} \rfloor$, and the average (counterclockwise) hop length is at most $\bar{L}$, so the number of contiguous counterclockwise hops required is

$$D_W \leq \left(P_{tot} - \left\lfloor \frac{P_{tot}N}{4\bar{L}} \right\rfloor \right) \cdot (N - L).$$

Applying the inequality $\lfloor x \rfloor > x - 1$ and proceeding,

$$\begin{aligned} D_W &< \left(P_{tot} - \frac{P_{tot}N}{4\bar{L}} + 1 \right) \cdot (N - \bar{L}) \\ &= \left(P_{tot} - \frac{P_{tot}N}{4\bar{L}} \right) \cdot (N - \bar{L}) + (N - \bar{L}) \\ &< \left(P_{tot} - \frac{P_{tot}N}{4\bar{L}} \right) \cdot (N - \bar{L}) + N \end{aligned}$$

where in the last line we used the fact that $\bar{L} > 0$. Next, to eliminate the dependence on $\bar{L}$, we would like to maximize the right-hand side over $\bar{L}$. To do this, we take the derivative with respect to $\bar{L}$ and set it to zero:

$$\begin{aligned} \left(\frac{P_{tot}N}{4\bar{L}^2} \right)(N - \bar{L}) - \left(P_{tot} - \frac{P_{tot}N}{4\bar{L}} \right) &= 0 \\ \frac{P_{tot}N^2}{4\bar{L}^2} - \frac{P_{tot}N}{4\bar{L}} - P_{tot} + \frac{P_{tot}N}{4\bar{L}} &= 0 \\ \frac{P_{tot}N^2}{4\bar{L}^2} - P_{tot} &= 0 \\ \bar{L}^2 &= \frac{N^2}{4} \\ \Rightarrow \bar{L} &= \frac{N}{2}. \end{aligned}$$

Knowing that the maximizing value of $\bar{L}$ is $N/2$, we substitute that value back into the original equation to obtain

$$\begin{aligned} D_W &< \left(P_{tot} - \frac{P_{tot}N}{4\left(\frac{N}{2}\right)} \right) \cdot \left(N - \left(\frac{N}{2} \right) \right) + N \\ &= \left(P_{tot} - \frac{P_{tot}}{2} \right) \left(\frac{N}{2} \right) + N \\ &= \frac{P_{tot}N}{4} + N. \end{aligned}$$

The total number of required wavelengths is then

$$\begin{aligned} \left\lceil \frac{D_W}{N} \right\rceil &= \left\lceil \frac{P_{tot}}{4} + 1 \right\rceil \\ &= \left\lceil \frac{P_{tot}}{4} \right\rceil + 1. \end{aligned}$$

Note that one additional wavelength is required to accommodate the counterclockwise set. However, if the original traffic set was unconnected and required the approach of Section II-C to turn it into a connected set (using an extra wavelength in the clockwise direction), it can share the same extra wavelength since the $\lceil PN/4 \rceil$ algorithm uses only an extra wavelength in the counterclockwise direction. In other words, for an unconnected general P-port traffic set, only a single extra wavelength is required, not two. ■

Here a total of at most $2\lceil(P_{tot}/4)\rceil$ converters are required. Each node i requires no more than P_i converters.

B. The $2\lceil PN/7\rceil$ Algorithm

Again we consider the case of a connected network. The network can be either symmetric or asymmetric; again let node i have P_i ports, and define $P_{tot} = \sum_{i=1}^{N} P_i$ to be the total number of calls in the system. Find the Euler trail and list the calls in adjacent order.

By dividing the calls into adjacent sets of 7, the results of Theorem 3 can be applied to route each set using at most 2 wavelengths. Therefore a total of $\lceil 2P_{tot}/7\rceil$. For a symmetric network, $P_{tot} = PN$, where P is the number of ports per node, and this number simplifies to $2\lceil PN/7\rceil$. For this reason, this slightly modified algorithm is called the $2\lceil PN/7\rceil$ algorithm.

For a connected network, a total of at most $\lfloor P_{tot}/7\rfloor$ converters are required. Again, in Section IV we will show how the wavelength assignment can be modified to distribute the converters nearly arbitrarily among all nodes. In particular, for symmetric networks, a modified wavelength assignment can be given that requires no more than $\lceil P/4\rceil + 1$ converters per node.

IV. The Converter-Shifting Algorithms

A. The Converter-Shifting Lemmas

In general, when a RWA algorithm gives a wavelength assignment for a traffic set, it will also specify the number of converters required at each node to support its wavelength assignment. However, this may result in inefficient use of converters since the network will have to be designed with the maximum number of converters (over all possible admissible traffic sets) at each node that the algorithm may require. For example, consider a 2-node network that sees one of two possible traffic sets, A and B. Suppose for a particular RWA traffic set A requires that node 1 have 3 converters and node 2 have 6, whereas in the RWA for traffic set B node 1 requires 6 and node 2 requires 3. Then if sets A and B are to be supported in a rearrangeably nonblocking manner, nodes 1 and 2 must both have $\max\{3,6\} = 6$ converters, for a total of 12 converters between them, even though at most 9 converters are ever used at any given time.

In this section we provide a procedure for modifying a given wavelength assignment so that the conversion requirement can be moved arbitrarily from any node to any other node while preserving the routing of the calls. If certain criteria are met, removing one converter from a given node will require the addition of only one converter at a different node. We call this a *one-to-one* exchange. Otherwise, removing one converter from a given node will require the addition of two converters at a different node; we call this a *one-to-two* exchange.

We first define some terminology that we will find useful. A wavelength converter, when in use, converts an input wavelength to a different output wavelength. Suppose two converters are operating in the same direction (either clockwise or counterclockwise). If the output wavelength of converter 1 is the same as the input wavelength of converter 2, then we say that converter 1 is *adjacent* to converter 2, and vice versa. In particular, converter 2 is *forward adjacent* to converter 1, and converter 1 is *backward adjacent* to converter 2. Converters cannot be adjacent if they are operating in different directions.

The next two lemmas give conditions under which converters can be moved from one node to another in a one-to-one exchange. The lemmas differ in the direction a converter is shifted relative to its adjacency to the destination.

Lemma 4: If for a given RWA a converter c_j at node j is forward adjacent to a converter c_i at node i, a modified wavelength assignment can be devised that does not require a converter at node i but may require an additional converter at node j.

Proof: Without loss of generality, suppose the converters are operating in the clockwise direction. Call the set of all links encountered travelling from i to j in the clockwise direction the *swap set*. Let the input and output wavelengths of c_i be λ_1 and λ_2, respectively. Let the output wavelength of c_j be λ_3.

Move all traffic in the swap set on wavelength λ_1 to λ_2, and move all traffic in the swap set previously on λ_2 to λ_1. Now c_i is no longer required, since the call coming into node i on λ_1 continues on λ_1 after the swap. Also notice that calls in the swap set on λ_1 must have started at or after node i. The input wavelength of c_j becomes λ_1 after the swap, since the call which previously had been coming in on λ_2 was moved to λ_1. The output wavelength of c_j remains the same.

There remains one loose end to tie up. There may previously have been a call which entered node j on λ_1 and continued out on λ_1. Since after the swap this call is now entering on λ_2, an additional converter is required to convert it to λ_1 for it to continue out on λ_1 as before. Note that if the call had terminated at node j, then this converter would not be needed. ■

Lemma 5: If for a given RWA a converter c_j at node j is forward adjacent to a converter c_i at node i, a modified wavelength assignment can be devised that does not require a converter at node j but may require an additional converter at node i.

Proof: The proof is very similar to the proof of Lemma 4. Call the set of all links encountered travelling from i to j in the clockwise direction the *swap set*. Let the input and output wavelengths of c_i be λ_1 and λ_2, respectively. Let the output wavelength of c_j be λ_3.

Move all traffic in the swap set on wavelength λ_3 to λ_2, and move all traffic in the swap set previously on λ_2 to λ_3. Now c_j is no longer required, since the call previously entering node j on λ_2 has been moved to λ_3, and may continue on λ_3 without needing a converter. The output wavelength of c_i becomes λ_3 after the swap, since the call which previously exited on λ_2 was moved to λ_3. The input wavelength of c_i remains the same.

Again there is a loose end to tie up. There may previously have been a call which entered node i on λ_3 and continued out on λ_3. Since after the swap this call is continuing on λ_2, an additional converter is required to convert it from λ_3 to λ_2. Note that if the call had started at node i, then again this converter would not be needed. ■

An example of a one-to-one exchange of the type described in Lemma 4 is shown in Figs. 8 and 9. Finally, we have a general theorem for shifting converters if no adjacent converter is available at the destination node.

Lemma 6: If for a given RWA there does not exist any converter at node j that is adjacent to any converter at node i, a modified wavelength assignment can be devised that requires

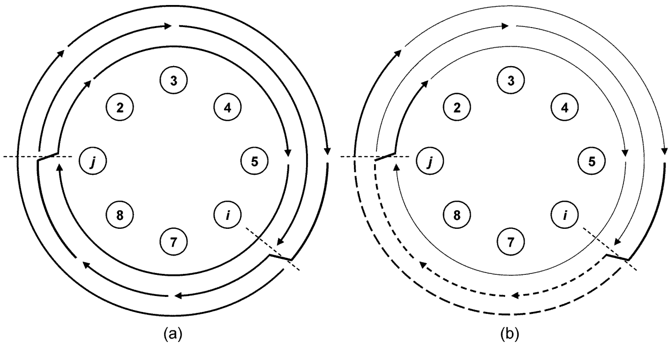


Fig. 8. (a) The original RWA of calls on the clockwise direction. Note that there is no requirement that the traffic set obey a P-port condition. Converters are used at nodes i and j. (b) The same ring, with related calls marked. Calls affected by the converter shifting are in bold, while unaffected calls are in light grey. The swap set consists of the dotted calls and parts of calls.

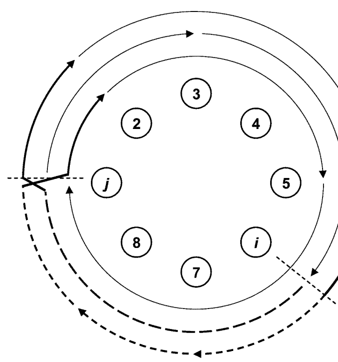


Fig. 9. All calls or parts of calls in the short dotted lines have exchanged wavelengths with those on the long dotted lines. Note that while a converter is no longer required at node i, an extra one is now being used at node j.

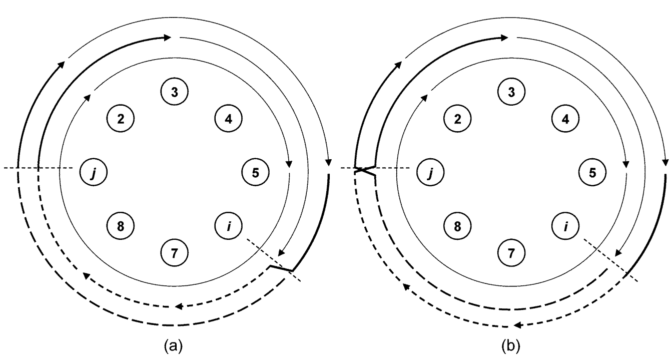


Fig. 10. (a) The original RWA of calls on the clockwise direction. A single converters is used by node i. Calls affected by the converter shifting are in bold, while unaffected calls are in light grey. The swap set consists of the dotted calls and parts of calls. (b) All calls or parts of calls in the short dotted lines have exchanged wavelengths with those on the long dotted lines. Note that while a converter is no longer required at node i, two are used at node j.

one less converter at i but may require up to two more converters at node j.

Proof: The proof is identical to the proof of Lemma 4, except that since there is no existing adjacent converter c_j to use at node j, a new one is required. ■

An example of one-to-two exchange is shown in Fig. 10. The proofs of the preceding lemmas provide an algorithm for shifting converters from node to node.

In the following two subsections, we use the converter-shifting lemmas to first describe a method for moving all converters to a single node (typically called the *hub*), then describe a method for distributing them arbitrarily among all nodes while requiring at most one additional converter per node. The techniques used in these two examples can then be applied in a straightforward manner to implement any other configurations of interest.

B. Applications to the $\lceil PN/4 \rceil$ Algorithm

In this section, we demonstrate the use of the converter-shifting lemmas on the $\lceil PN/4 \rceil$ algorithm to create two interesting network architectures, the hub architecture and the symmetric node architecture.

1) Hub Architecture: It may be desirable to concentrate all converters at a single node, called the hub. This can be done using the converter-shifting lemmas to move all converters to the hub at a cost of at most two additional converters.

Recall that by construction at most $\lceil PN/4 \rceil - 1$ converters are used in each direction. Consider first the clockwise direction. Since by construction the converters can be traversed in adjacent order, without loss of generality we may index the converters so that converter c_i has input wavelength λ_i and output wavelength λ_{i+1}, for $i = 1, \ldots, \lceil PN/4 \rceil - 2$.

Suppose node n_h is chosen to be the hub node. According to Lemma 6, we can move c_1 to node n_h using a one-to-two exchange. Next, move converter c_2 to node n_h. Since by choice of indexing the input wavelength of c_2 is the output wavelength of c_1, by Lemma 5 it can be moved using a one-to-one exchange. Iterating through the rest of the converters, the same argument can be applied to perform one-to-one exchanges. After all exchanges are complete, there are a total of $\lceil PN/4 \rceil$ converters at the hub – one more than the previous total, due to the initial one-to-two exchange.

The same procedure can be repeated for the counterclockwise direction, resulting in an additional $\lceil PN/4 \rceil$ converters being collected at the hub. After this procedure, all conversion is now concentrated at the hub, which requires $\lceil PN/2 \rceil$ converters.

2) Symmetric Node Architecture: In other cases, we may prefer to have each node have the same number of converters. Again, this can be accomplished by using the converter-shifting lemmas to move the converters such that each node has no more than $\lceil P/2 \rceil + 1$ converters.

The procedure is as follows: first, apply the method of the previous section to create a hub architecture. There are now $\lceil PN/4 \rceil$ adjacent converters at the hub in either direction. Divide the remaining $N - 1$ nodes into two sets of equal size (N odd). Call one set the *clockwise set*, and the other the *counterclockwise set*. First consider the clockwise direction. Move $\lceil P/2 \rceil$ of the converters in adjacent order to one of the $(N-1)/2$ nodes in the clockwise set. The first requires a one-to-two exchange, while all remaining converters are moved one-to-one. This places $\lceil P/2 \rceil + 1$ converters at that node. Repeat with all remaining nodes in the clockwise set. At the end of the procedure, all nodes in the clockwise set have $\lceil P/2 \rceil + 1$ converters in the clockwise direction.

Repeat this procedure with the counterclockwise set using the counterclockwise converters. This leaves all nodes in the counterclockwise set with $\lceil P/2 \rceil + 1$ converters in the counterclockwise direction. The hub itself has a total of $\lceil P/2 \rceil$ converters, half in either direction. Thus no node requires more than $\lceil P/2 \rceil + 1$ converters.

Finally, recall that the original algorithm required no more than P converters at any given node. We always retain the option of not doing any converter shifting if $\lceil P/2 \rceil + 1 > P$. (As a side note, we point out that the only time this occurs is at $P = 1$.) Therefore the final result is that the number of converters required per node is given as $\min\{\lceil P/2 \rceil + 1, P\}$.

C. Applications to the $2\lceil PN/7 \rceil$ Algorithm

In this section, we demonstrate the use of the converter-shifting lemmas on the $2\lceil PN/7 \rceil$ algorithm to again create a hub and symmetric node architectures.

1) Hub Architecture: The converter-shifting lemmas can be used to move all converters to a single node. For the $2\lceil PN/7 \rceil$ algorithm, converter adjacency is not guaranteed, and hence redistribution requires one-to-two exchanges. Hence the hub has at most $2\lfloor PN/7 \rfloor$ converters.

2) Symmetric Node Architecture: The converter-shifting lemmas can also be used to move converter requirements to ensure that each node requires no more than $\lceil P/4 \rceil$ converters.

The procedure is as follows. Locate the nodes which require more than $\lceil P/4 \rceil$ converters. Define these nodes to be members of the set R requiring relocation. Consider the first converter in the set R. Locate a node not contained in R which currently has fewer than $\lceil P/4 \rceil$ converters, and move it to that node. We call this the *relocation step*, which is at worst a one-to-two exchange. Repeat the relocation step until the number of converters at that node drops to $\lceil P/4 \rceil$. Remove that node from the set R, then move onto the next node in R and repeat, until the set R is empty.

We claim that we can always perform the relocation step for all nodes in R; that is, we never run out of nodes with fewer than $\lceil P/4 \rceil$ converters while there remain nodes in R with converters which need to be relocated. This claim is formalized in the following theorem.

Theorem 6: Define the *excess demand* D for converters to be the sum of the minimum number of converters which need to be removed from each node so that the number of the converters at the node does not exceed $\lceil P/4 \rceil$. Define the *excess capacity* C to be the sum of the maximum number of converters which could be added at each node without exceeding $\lceil P/4 \rceil$. Denote by X_i the quantity of converters required at node i by a given RWA. Mathematically, these quantities are related by:

$$\begin{aligned} D &= \sum_{i=1}^{N} \max\left(X_i - \left\lceil \frac{P}{4} \right\rceil, 0\right) \\ &= \sum_{i \in R} \left(X_i - \left\lceil \frac{P}{4} \right\rceil\right) \\ C &= \sum_{i=1}^{N} \max\left(\left\lceil \frac{P}{4} \right\rceil - X_i, 0\right) \\ &= \sum_{i \in R^C} \left(\left\lceil \frac{P}{4} \right\rceil - X_i\right) \end{aligned}$$

where R^C denotes the complement of R; i.e., R^C is composed of those nodes not contained in R.

Then the theorem asserts that

$$2D \leq C.$$

Proof: Index the nodes $n_1, \ldots, n_N$ such that $n_1, \ldots, n_j$ all have more than $\lceil P/4 \rceil$ converters, while the remaining nodes $n_{j+1}, \ldots, n_N$ do not. By this choice of indexing, the set R is composed of the nodes $\{n_1, \ldots, n_j\}$. The expressions for D and C can be written as

$$\begin{aligned} D &= \sum_{i=1}^{j} \left(X_i - \left\lceil \frac{P}{4} \right\rceil\right) \\ &= \left(\sum_{i=1}^{j} X_i\right) - j\left\lceil \frac{P}{4} \right\rceil \\ C &= \sum_{i=j+1}^{N} \left(\left\lceil \frac{P}{4} \right\rceil - X_i\right) \\ &= (N-j)\left\lceil \frac{P}{4} \right\rceil - \left(\sum_{i=j+1}^{N} X_i\right). \end{aligned}$$

To prove the theorem, we must show that $C - 2D \geq 0$. To see this, begin with

$$\begin{aligned} C - 2D &= (C - D) - D \\ &= (N-j)\left\lceil \frac{P}{4} \right\rceil - \sum_{i=j+1}^{N} X_i \\ &\quad - \sum_{i=1}^{j} X_i + j\left\lceil \frac{P}{4} \right\rceil - D \\ &= \left(N\left\lceil \frac{P}{4} \right\rceil - \sum_{i=1}^{N} X_i\right) - D \\ &\geq \left(N\left\lceil \frac{P}{4} \right\rceil - \left\lfloor \frac{PN}{7} \right\rfloor\right) - D \\ &\geq \left(\frac{PN}{4} - \frac{PN}{7}\right) - D \\ &= \frac{3}{28}PN - D \end{aligned}$$

where the first inequality arises from the fact that the total number of converters required $\sum_{i=1}^{N} X_i \leq \lfloor PN/7 \rfloor$, and the second is from the removal of the floor and ceiling functions.

We next need to determine an upper bound on the excess demand D. To develop this bound, we formulate an equivalent problem involving balls and jars. Consider the problem of distributing $\lfloor PN/7 \rfloor$ balls into N jars, where each jar can hold at most P balls stacked vertically, in order to maximize the total number of balls in the jars exceeding a height of $\lceil P/4 \rceil$. This is illustrated in Fig. 11. The balls correspond to converters, the jars to nodes, and the number of balls which exceed height $\lceil P/4 \rceil$ is equal to the quantity D.

An algorithm for maximizing the number of balls placed which exceed height $\lceil P/4 \rceil$ is to begin at the first jar, fill it with as many balls as possible, move to the next jar, and repeat. Then the number of jars required is $\lfloor PN/7 \rfloor / P \leq (PN/7)/P = N/7$, and each jar has an excess capacity of at most $P - \lceil P/4 \rceil \leq P - P/4 = 3P/4$. Therefore the excess demand is at most $D \leq (P/7)(3P/4) = 3PN/28$.

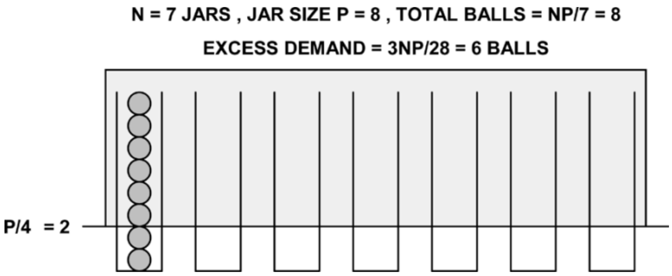


Fig. 11. An example of the ball distribution problem. The excess capacity (represented by balls falling in the shaded area) is maximized by filling each jar as much as possible before moving onto the next jar.

Using this inequality in (1), we then have

$$C - 2D \geq \frac{3}{28}PN - \frac{3}{28}PN$$
$$= 0$$

which proves the theorem. ■

A direct corollary of this theorem is that converters can be equally distributed so that no node needs more than $\lceil P/4 \rceil + 1$ converters. The one-to-two shifting is the reason for the extra "+1" term. In the worst case, it is possible that a converter may be shifted to a node outside the set R which prior to the shifting had $\lceil P/4 \rceil - 1$ converters; in this case, adding two additional converters gives it a total of $\lceil P/4 \rceil + 1$.

Again, since the original algorithm required no more than P converters per node, we retain the option of doing no shifting if $\lceil P/4 \rceil + 1 > P$. Therefore in the final assessment the number of converters required per node is $\min\{\lceil P/4 \rceil + 1, P\}$.

V. Conclusions

We considered the problem of implementing all virtual topologies on an N-node P-port network in a rearrangeably nonblocking fashion while trying to minimize the number of wavelengths and converters required. We show that for symmetric P-port networks, a lower bound on the number of wavelengths is $\lceil PN/4 \rceil$. We present an algorithm which achieves this lower bound by using $\lceil PN/4 \rceil$ wavelengths for connected topologies while using a total of no more than $\lceil PN/2 \rceil - 2$ converters. We also present a second algorithm which uses $2\lceil PN/7 \rceil$ wavelengths but requires fewer converters, a total of no more than $\lfloor PN/7 \rfloor$. The first algorithm achieves the minimum number of wavelengths required, while the second uses more wavelengths but greatly reduces the number of converters used. We also show how to turn the problem of implementing an unconnected traffic set into a modified problem of implementing a connected set by using a single additional wavelength. We then extend the results to general P-port networks, where we allow the number of ports P_i at each node i to vary, and show that for such networks the $\lceil PN/4 \rceil$ algorithm requires no more than $\lceil \sum_i P_i/4 \rceil + 1$ wavelengths for connected and unconnected traffic sets. A similar extension for the $\lceil PN/7 \rceil$ algorithm shows that it requires only $\lceil \sum_i P_i/7 \rceil + 1$ wavelengths.

Finally, we demonstrate a method for changing wavelength assignments to move converters arbitrarily from one node to another. If certain conditions are met, we show that this exchange is one-to-one; otherwise, the exchange is one-to-two. We also show how to apply this method to both the $\lceil PN/4 \rceil$ and $2\lceil PN/7 \rceil$ algorithms. For symmetric P-port networks, we demonstrate a hub topology for the $\lceil PN/4 \rceil$ algorithm which uses $\lceil PN/2 \rceil$ converters at the hub and no converters elsewhere, and a symmetric node topology which uses $\lceil P/2 \rceil + 1$ converters at each node. We also give a hub topology for the $2\lceil PN/7 \rceil$ algorithm which uses $2\lceil PN/7 \rceil$ converters at the hub and no converters elsewhere, and a symmetric node topology which uses at most $\lceil P/4 \rceil + 1$ converters at each node. For asymmetric networks, the expressions are the same except that $P_{tot} = \sum_i P_i$ replaces PN.

It is worth comparing the worst-case wavelength requirement to the wavelength requirement for static and uniform all-to-all traffic. In all-to-all uniform traffic, each node communicates with every other node. For N odd, this requires $(N^2 - 1)/8$ wavelengths [20], [21]. In our terminology, all-to-all traffic belongs to the admissible set of an N-node network with $N - 1$ ports, which have a worst-case bound of $N(N-1)/4$ wavelengths. Thus designing a network to support $P = N - 1$ calls per node uses twice as many wavelengths as a uniform all-to-all design. However, the P-port traffic model provides significantly more flexibility than the uniform all-to-all model. Furthermore, an argument given in [1] can be used to show that a large number of topologies require the lower bound of $\lceil PN/4 \rceil$ wavelengths for the P-port case, showing that this bound is not inflated to support only a small number of worst-case scenarios.

Appendix

In this section we consider the number of wavelengths required by the $\lceil N/4 \rceil$ algorithm in the counterclockwise direction for the case of $k = \lfloor N^2/4\bar{L} \rfloor$. Recall that the number of hops of traffic in the counterclockwise set was given by

$$\begin{aligned} D_W &= (N-k) \cdot (N - \hat{L}) \\ &\leq (N-k) \cdot (N - \bar{L}) \\ &= \left(N - \left\lfloor \frac{N^2}{4\bar{L}} \right\rfloor\right) \cdot (N - \bar{L}). \end{aligned}$$

Consider the maximization of the right-hand side; that is, the function

$$f(L) = \left(N - \left\lfloor \frac{N^2}{4L} \right\rfloor\right) \cdot (N - L). \quad (1)$$

The number of nodes N must obviously be an integer, and we can also deduce that the average hop length $\bar{L}$ is also integer. To see this, recall that we assumed the traffic set was connected. This implies that, starting at any node, we can proceed in adjacent order through all the calls in the clockwise direction and return to the same node. Thus, the total number of hops of traffic in the clockwise direction must be an integer multiple of N. Therefore the average hop length, which we obtain by dividing the total hop length by the number of nodes N, must also be integer.

For the proof we will also only consider the case where N is even. There is no loss of generality because in all cases of practical interest, this assumption holds. To see this, consider a ring network with N odd. We can add a fictitious node $N + 1$ to make the total number of nodes even. We alter the traffic set by arbitrarily picking any call from the original traffic set.

Suppose this call is from node n_i to n_j, denoted by (n_i, n_j). We remove this call from the traffic set and replace it by two calls (n_i, x) and (x, n_j). Observe that this new traffic set, over the $(N+1)$-node ring, is now a maximal single-port traffic set. It also retains connectedness.

The number of wavelengths required to route the new traffic set using the $\lceil N/4 \rceil$ algorithm is $\lceil (N+1)/4 \rceil$. Since for N odd $\lceil (N+1)/4 \rceil = \lceil N/4 \rceil$, no additional wavelengths are required by this procedure. Once routes have been found for all calls, remove the fictitious node x. Then use the route determined for the calls (n_i, x) and (x, n_j) to route the original call (n_i, n_j). This shows that it is sufficient to consider the case of only N even, because it allows us to also perform RWA for N odd without using any additional wavelengths.

Returning our attention to the function $f(L)$, we are interested in finding an upper bound. The goal will be to show that the total hops of traffic is no greater than $N^2/4$, and by combining this with the fact that each wavelength provides N hops of traffic capacity, we will also prove that the counterclockwise set requires no more than $\lceil N/4 \rceil$ wavelengths.

The proof will proceed by showing the following two relations:

1) For all $k \in \{1, \ldots, N/2\}$, $f(N/2) \geq f((N/2) - k)$
2) For all $k \in \{1, \ldots, N/2\}$, $f(N/2) \geq f((N/2) + k)$

Together, the two relations show that $f(L)$ is maximized at $L = N/2$. Since $f(N/2) = N^2/4$, this leads to the desired result.

We proceed with showing the first inequality. We first introduce a useful lemma, followed by the proof of the theorem.

Lemma 7: For $k \in \{1, \ldots, N/2\}$ and $N/2$ integer,

$$\left\lfloor \frac{N^2}{4\left(\frac{N}{2} - k\right)} \right\rfloor \geq \frac{N}{2} + k.$$

Proof: We begin by showing

$$\frac{N^2}{4\left(\frac{N}{2} - k\right)} = \frac{N^2}{2N\left(1 - \frac{2k}{N}\right)} = \frac{N}{2\left(1 - \frac{2k}{N}\right)}.$$

Using this result, we can then also show that

$$\frac{N^2}{4\left(\frac{N}{2} - k\right)} - \frac{N}{2} = \frac{N}{2\left(1 - \frac{2k}{N}\right)} - \frac{N}{2} = \frac{k}{1 - \frac{2k}{N}} > k$$

and therefore

$$\frac{N^2}{4\left(\frac{N}{2} - k\right)} > \frac{N}{2} + k.$$

Taking the floor of both sides

$$\left\lfloor \frac{N^2}{4\left(\frac{N}{2} - k\right)} \right\rfloor \geq \left\lfloor \frac{N}{2} + k \right\rfloor = \frac{N}{2} + k$$

where the last step follows from the fact that both $N/2$ and k are integers. This proves the lemma. ∎

Theorem 7: For $k \in \{1, \ldots, N/2\}$ and $N/2$ integer,

$$f\left(\frac{N}{2} - k\right) \leq f\left(\frac{N}{2}\right).$$

Proof: Beginning at the definition of $f(L)$, we have:

$$\begin{aligned} f\left(\frac{N}{2} - k\right) &= \left(N - \left\lfloor \frac{N^2}{4\left(\frac{N}{2} - k\right)} \right\rfloor\right) \cdot \left(N - \left(\frac{N}{2} - k\right)\right) \\ &\leq \left(N - \left(\frac{N}{2} + k\right)\right) \cdot \left(\frac{N}{2} + k\right) \end{aligned}$$

where the last inequality was obtained using Lemma 7. Continuing, a few additional algebraic steps gives us

$$\begin{aligned} f\left(\frac{N}{2} - k\right) &\leq \left(\frac{N}{2} - k\right) \cdot \left(\frac{N}{2} + k\right) \\ &= \frac{N^2}{4} - k^2 \\ &\leq \frac{N^2}{4}. \end{aligned}$$

Since $f(N/2) = N^2/4$, this shows that

$$f\left(\frac{N}{2} - k\right) \leq f\left(\frac{N}{2}\right)$$

which proves the theorem. ∎

The proof of the second inequality parallels the development of the proof of the first very closely. Again, a helpful lemma will first be developed before the theorem is presented.

Lemma 8: For $k \in \{1, \ldots, N/2\}$ and $N/2$ integer,

$$\left\lfloor \frac{N^2}{4\left(\frac{N}{2} + k\right)} \right\rfloor \geq \frac{N}{2} - k.$$

Proof: We begin by observing

$$\frac{N^2}{4\left(\frac{N}{2} + k\right)} = \frac{N^2}{2N\left(1 + \frac{2k}{N}\right)} = \frac{N}{2\left(1 + \frac{2k}{N}\right)}.$$

Using the above, we have

$$\begin{aligned} \frac{N}{2} - \frac{N^2}{4\left(\frac{N}{2} + k\right)} &= \frac{k}{1 + \frac{2k}{N}} \\ &< k \\ \Rightarrow \frac{N^2}{4\left(\frac{N}{2} + k\right)} &> \frac{N}{2} - k. \end{aligned}$$

Taking the floor of both sides,

$$\left\lfloor \frac{N^2}{4\left(\frac{N}{2} + k\right)} \right\rfloor \geq \left\lfloor \frac{N}{2} - k \right\rfloor = \frac{N}{2} - k$$

where the last line follows from the fact that both $N/2$ and k are integers. This proves the lemma. ∎

Theorem 8: For $k \in \{1, \ldots, N/2\}$ and $N/2$ integer,

$$f\left(\frac{N}{2} + k\right) \leq f\left(\frac{N}{2}\right).$$

Proof: Beginning at the definition of $f(L)$ and applying Lemma 8, we have:

$$\begin{aligned} f\left(\frac{N}{2}+k\right) &\leq \left(N-\left(\frac{N}{2}-k\right)\right)\cdot\left(\frac{N}{2}-k\right) \\ &\leq \frac{N^2}{4}-k^2 \\ &\leq \frac{N^2}{4}. \end{aligned}$$

Since $f(N/2) = N^2/4$, this shows that

$$f\left(\frac{N}{2}+k\right) \leq f\left(\frac{N}{2}\right)$$

which proves the theorem. ■

References

[1] A. Narula-Tam, P. J. Lin, and E. Modiano, "Efficient routing and wavelength assignment for reconfigurable WDM networks," *IEEE J. Select. Areas Commun.*, vol. 20, no. 1, pp. 75–88, Jan. 2002.

[2] R. Ramaswami and K. N. Sivarajan, *Optical Networks: A Practical Perspective*. San Mateo, CA: Morgan Kaufmann, 1998.

[3] I. Chlamtac, A. Ganz, and G. Karmi, "Lightpath communications: An approach to high bandwidth optical WANs," *IEEE Trans. Commun.*, vol. 40, no. 7, pp. 1171–1182, Jul. 1992.

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

[5] R. M. Krishnaswamy and K. N. Sivarajan, "Design of logical topologies: A linear formulation for wavelength-routed optical networks with no wavelength changers," *IEEE/ACM Trans. Networking*, vol. 9, no. 2, pp. 186–198, Apr. 2001.

[6] R. Ramaswami and K. N. Sivarajan, "Routing and wavelength assignment in all-optical networks," *IEEE/ACM Trans. Networking*, vol. 3, no. 5, pp. 489–500, Oct. 1995.

[7] A. Birman, "Computing approximate blocking probabilities for a class of all-optical networks," *IEEE J. Select. Areas Commun.*, vol. 14, no. 6, pp. 852–857, Jun. 1996.

[8] R. A. Barry and P. A. Humblet, "Models of blocking probability in all-optical networks with and without wavelength changers," *IEEE J. Select. Areas Commun.*, vol. 14, no. 6, pp. 858–867, Jun. 1996.

[9] M. Kovacevic and A. Acampora, "Benefits of wavelength translation in all-optical clear-channel networks," *IEEE J. Select. Areas Commun.*, vol. 14, no. 6, pp. 868–880, Jun. 1996.

[10] S. Subramaniam, M. Azizoglu, and A. Somani, "A new analytical model for multifiber WDM networks," *IEEE J. Select. Areas Commun.*, vol. 18, no. 10, pp. 2138–2145, Oct. 2000.

[11] Y. Zho, G. N. Rouskas, and H. G. Perros, "A path decomposition algorithm for computing blocking probabilities in wavelength routing networks," *IEEE/ACM Trans. Networking*, vol. 8, no. 6, pp. 747–762, Dec. 2000.

[12] S. Ramesh, G. N. Rouskas, and H. G. Perros, "Computing blocking probabilities in multiclass wavelength-routing networks with multicast calls," *IEEE J. Select. Areas Commun.*, vol. 20, no. 1, pp. 89–96, Jan. 2002.

[13] E. Karasan and E. Ayanoglu, "Effects of wavelength routing and selection algorithms on wavelength conversion gain in WDM optical networks," *IEEE/ACM Trans. Networking*, vol. 6, no. 2, pp. 186–196, Apr. 1998.

[14] L. Li and A. K. Somani, "Dynamic wavelength routing using congestion and neighborhood information," *IEEE/ACM Trans. Networking*, vol. 7, no. 5, pp. 779–786, Oct. 1999.

[15] O. Gerstel, G. Sasaki, S. Kutten, and R. Ramaswami, "Worst-case analysis of dyanmic wavelength allocation in optical networks," *IEEE/ACM Trans. Networking*, vol. 7, no. 6, pp. 833–845, Dec. 1999.

[16] R. Ramaswami and G. Sasaki, "Multiwavelength optical networks with limited wavelength conversion," *IEEE/ACM Trans. Networking*, vol. 6, no. 6, pp. 744–754, Dec. 1998.

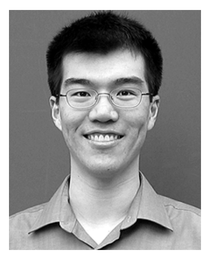
[17] P. Saengudomlert, E. Modiano, and R. G. Gallager, "An on-line routing and wavelength assignment algorithm for dynamic traffic in a WDM bidirectional ring," presented at the JCIS Joint Conf. Information Sciences, Mar. 2002.

[18] ——, "On-line routing and wavelength assignment for dynamic traffic in WDM ring and torus networks," presented at the IEEE INFOCOM, San Francisco, CA, Apr. 2003.

[19] M. N. S. Swamy and K. Thulasiraman, *Graphs, Networks, and Algorithms*. New York: Wiley, 1981.

[20] J. M. Simmons, E. L. Goldstein, and A. A. M. Saleh, "Quantifying the benefit of wavelength add-drop in WDM rings with distance-independent and dependent traffic," *J. Lightwave Technol.*, vol. 17, no. 1, pp. 48–57, Jan. 1999.

[21] A. F. Elrafaie, "Multiwavelength survivable ring network architectures," in *Proc. ICC*, May 1993, pp. 1245–1251.



Li-Wei Chen (S'97) received the B.A.Sc. degree in electrical and computer engineering from Queen's University, Kingston, ON, Canada, in 1998, and the M.A.Sc. degree in electrical and computer engineering from the University of Toronto, Toronto, ON, Canada, in 2000. He is currently working toward the Ph.D. degree in the Laboratory for Information and Decision Systems (LIDS) at the Massachusetts Institute of Technology (MIT), Cambridge.

His research interests are in the area of network architecture, with an emphasis on optical and high-speed networks.

Eytan Modiano (S'90–M'93–SM'00) received the B.S. degree in electrical engineering and computer science from the University of Connecticut, Storrs, in 1986, and the M.S. and Ph.D. degrees, both in electrical engineering, from the University of Maryland, College Park, in 1989 and 1992, respectively.

He was a Naval Research Laboratory Fellow between 1987 and 1992 and a National Research Council Post Doctoral Fellow during 1992–1993 while he was conducting research on security and performance issues in distributed network protocols. Between 1993 and 1999, he was with the Communications Division at MIT Lincoln Laboratory where he designed communication protocols for satellite, wireless, and optical networks and was the project leader for MIT Lincoln Laboratory's Next Generation Internet (NGI) project. He joined the MIT faculty in 1999, where he is presently an Associate Professor in the Department of Aeronautics and Astronautics and the Laboratory for Information and Decision Systems (LIDS). His research is on communication networks and protocols with emphasis on satellite, wireless, and optical networks.