

Efficient Routing and Wavelength Assignment for Reconfigurable WDM Networks with Wavelength Converters

Li-Wei Chen, Eytan Modiano

Abstract—We consider the problem of wavelength assignment in a reconfigurable bi-directional ring network 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 virtual topologies in a rearrangeably non-blocking fashion, and provide an algorithm that meets this bound for connected topologies using no more than $\lceil PN/2 \rceil$ wavelength converters. This improves over the tight lower bound of $\lfloor PN/3 \rfloor$ wavelengths required for such rings given in [1] if no wavelength conversion is available. We also provide another algorithm that uses more wavelengths yet requires significantly fewer converters. Both algorithms are then extended to the case of unconnected topologies using at most one additional wavelength. Finally, we 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 $\lceil P/2 \rceil$ converters at each node.

Index Terms—Graph theory, combinatorics

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.

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 matrix of 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.

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

This work was supported by the National Science Foundation (NSF) under Grant ANI-0073730.

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 modelling 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], [8], [9], [10], [11], [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 non-blocking*; algorithms which require call rearrangement are called *rearrangeably non-blocking*. 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 $\lfloor PN/3 \rfloor$ wavelengths is required to support the worst-case traffic set if no wavelength conversion is employed. Moreover, in [1] a rearrangeably non-blocking RWA algorithm is provided which achieved this bound. An online extension based on these ideas was presented in [17] which additionally attempts to minimize the number of rearrangements required. 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 non-blocking 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. 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 has P transmitters and receivers, and hence can transmit and receive P different calls. 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 possible traffic set in a P -port network in a rearrangeably non-blocking 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: 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. A result from [1] is also applied to show that a significant number of topologies require as many wavelengths as this lower bound (i.e. the number is not artificially high due to a negligibly small number of pathological cases). 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 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 non-blocking fashion. Our initial goal is to design a RWA algorithm which minimize 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 $\{c_1, c_2, \dots, c_N\}$ 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 $k = \lfloor N^2/4\bar{L} \rfloor$. Find a set of k adjacent calls with average clockwise hop length \bar{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.

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 $k = \lfloor N^2/4\bar{L} \rfloor = \lfloor 16/3 \rfloor = 5$. Choose the clockwise set to be

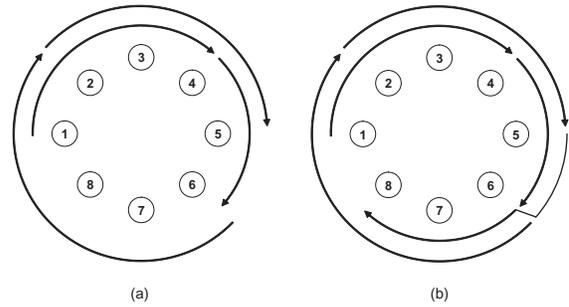


Fig. 1. (a) The routing and wavelength assignment of calls in the clockwise set after the forward pass. The inner arrows represent calls on λ_1 , the outer arrows are calls on λ_2 . (b) The complete RWA on the clockwise direction after the backward pass.

the set of calls $\{(1,4), (4,6), (6,2), (2,5), (5,8)\}$, with average hop length $\bar{L} = (3 + 2 + 4 + 3 + 3)/5 = 3 \leq \bar{L}$. The counterclockwise set then consists of the remaining calls, $\{(8,3), (3,7), (7,1)\}$. Note that the average hop length obeys $\bar{L} = (3 + 4 + 2)/3 \geq \bar{L}$ 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 Figure 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 Figure 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 $k = \lfloor N^2/4\bar{L} \rfloor$ 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 $\lceil N/2 \rceil$ 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 \bar{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

$$\begin{aligned} \frac{1}{n} \cdot (L_1 + \dots + L_n) &> \bar{L} \\ \frac{1}{n} \cdot (L_2 + \dots + L_{n+1}) &> \bar{L} \\ &\dots \\ \frac{1}{n} \cdot (L_{N-n+2} + \dots + L_N + L_1) &> \bar{L} \\ &\dots \\ \frac{1}{n} \cdot (L_N + L_1 + \dots + L_{n-1}) &> \bar{L} \end{aligned}$$

Summing the entire set of N inequalities, we obtain

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

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 $\frac{1}{n}$. Equivalently,

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

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 \bar{L} , we have that

$$\begin{aligned} \hat{L} &= L_{n+1} + \dots + L_N < (N - n)\bar{L} \\ \bar{L} &= L_1 + \dots + L_n \leq n\bar{L} \end{aligned}$$

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

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

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 $n = k = \left\lfloor \frac{N^2}{4L} \right\rfloor$ 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 $\lceil N/2 \rceil$ 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 calls can be added to increase $N/4$ to the nearest integer.) First note that since $N/4$ 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 $\bar{L} \leq \bar{L}$. Then the total number of hops required to accommodate the clockwise set, denoted by D_C , is

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

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. Denote the average clockwise hop length \hat{L} , and recall from Lemma 1 that $\hat{L} \geq \bar{L}$. Therefore the average counterclockwise hop length must be $N - \hat{L}$. Denote the total number of contiguous hops required to accommodate the counterclockwise set by D_W . Then,

$$D_W = (N - k) \cdot (N - \hat{L})$$

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

It can be shown that for integer N and \bar{L} , the last quantity is maximized at $\bar{L} = N/2$, giving us

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

which also fits in the $N/4$ wavelengths. It remains only to show that \bar{L} must also be an integer. To see this, note that if the traffic set is connected, it must form a single cycle; i.e. when all calls are indexed in adjacent order, the source node of the first call must be the destination node of the last call. Therefore when all calls are routed in the clockwise direction, the traffic set must form an integer number of cycles around the ring. Therefore the total length $\sum_i L_i$ must be an integer multiple of N . Therefore the average hop length $\bar{L} = \frac{1}{N} \sum_i L_i$ must be an integer.

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.

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.

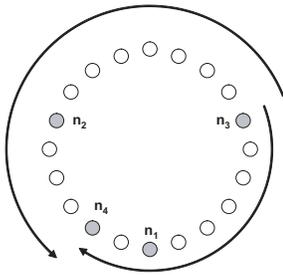


Fig. 2. Beginning at node n_3 , since we first encounter node n_1 before n_4 when travelling in the clockwise direction, we must encounter n_4 before n_1 when travelling in the counterclockwise direction.

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 Figure 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 second can fit alone on a second wavelength. ■

Figures 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

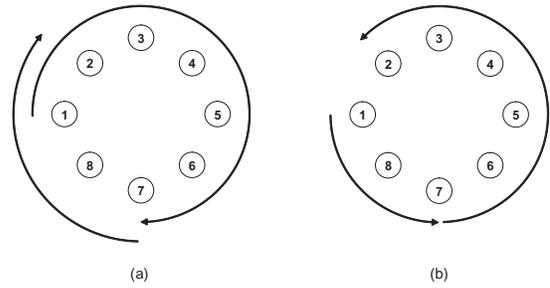


Fig. 3. (a) This adjacent pair cannot be placed on a single wavelength in the clockwise direction. (b) Therefore by Lemma 2, it can fit without converters on a single wavelength in the counterclockwise direction.

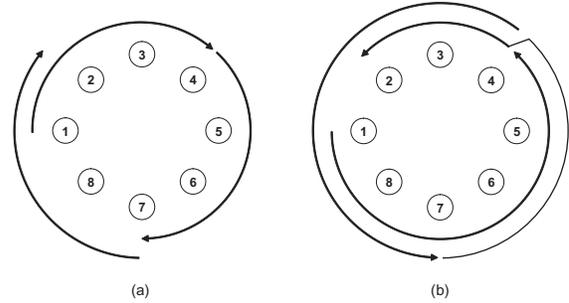


Fig. 4. (a) The adjacent triplet (n_1, n_4) , (n_4, n_7) , (n_7, n_2) cannot be placed on a single wavelength in the clockwise direction. (b) Therefore by Lemma 3, it can fit on two wavelengths in the counterclockwise direction using only a single converter. The converter is required at node 4 in this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

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 call 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 , $1 \leq j \leq c$. 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: By construction, it uses at most

$2\lceil N/7 \rceil$ wavelengths. Furthermore, since each set of 7 uses at most a single converter, the total number of converters required is upper-bounded by $\lceil N/7 \rceil$. 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 $\lceil (N/7) \rceil$ 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 non-maximal 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.

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, \dots, 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 $d_1,$

\dots, d_c are in *clockwise order*; i.e. after renumbering, travelling clockwise 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 $(d_i, d_{i \oplus 1})$ for $1 \leq i \leq c$. 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 $(d_c, d_{c-1}), (d_{c-1}, d_{c-2}), \dots, (d_3, d_2), (d_2, d_1)$. Note that this traffic set simply traverses all the d_i 's in descending order. Since the d_i 's were chosen in clockwise order by ascending i , it follows that they must be in counterclockwise order by descending i . Therefore all calls in T_1 can be fit onto a single wavelength in the counterclockwise 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 clockwise order and we do not need to renumber the cycles.

Following the preceding approach, in the superset call (1,4) becomes (1,3). Similarly, (8,3) becomes (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 $T_R = \{(3,4), (4,3)\}$.

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 Figures 5 and 6 respectively. Finally, the calls that were split during the creation of T_S and T_R are reconnected

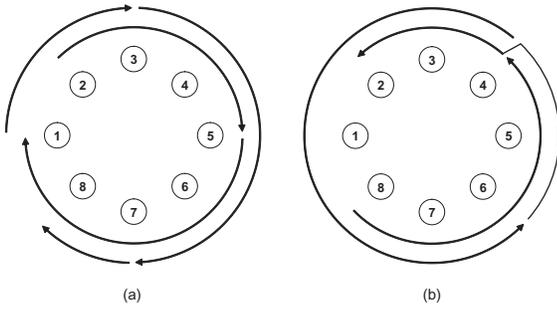


Fig. 5. The RWA for superset T_S of Example 2.

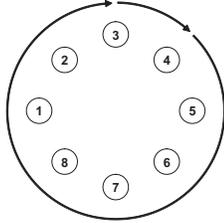


Fig. 6. The RWA of residual set T_R .

using wavelength converters in Figure 7.

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

III. MULTI-PORT 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.

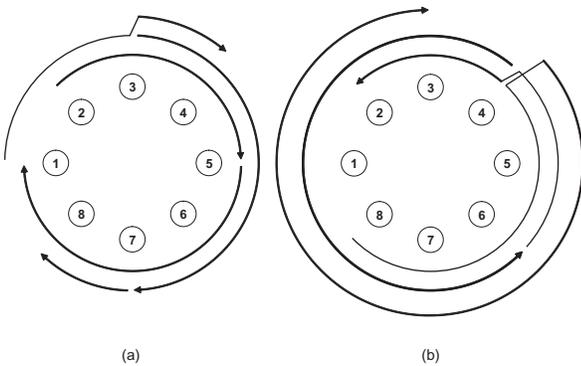


Fig. 7. (a) and (b) show the final RWA for Example 2 in the clockwise and counterclockwise directions, respectively. Note that although the call (8,3) in (b) ended up being routed partly in the counterclockwise direction and partly in the clockwise direction, the hops in the clockwise direction do not 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.

If the logical topology is connected, then the directed graph contains a directed Euler trail [18] 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 $k = \lfloor PN^2/4\bar{L} \rfloor$. Find a set of k adjacent calls with average clockwise hop length \bar{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 $\lceil (P/2) + 1 \rceil$ 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) *Asymmetric Multi-Port Networks:* We next consider asymmetric networks where each node i has P_i ports, and is able to transmit and receive at most P_i calls. Let $P_{tot} = \sum_{i=1}^N P_i$ be the total number of calls in the system. For a connected topology, we can find the Euler trail and obtain a sequence of P_{tot} adjacent calls.

It can be shown that the proof of the fact that only $\lceil PN/4 \rceil$ wavelengths were required in the symmetric port case relied only on PN being an integer multiple of N . Therefore the results still hold if P_{tot} is an integer multiple of N . This leads to the following theorem.

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

Here a total of at most $2\lceil (P_{tot}/4) \rceil - 2$ converters are required, and each node i requires no more than P_i converters if the traffic set is connected.

The approach of Section II-C can be used for unconnected networks, with some minor modification to the cycle formation. At the end of Step 4, it is possible that the same node may be contained in multiple cycles. Therefore a final step, Step 5,

is required where any two cycles which contain the same node should be merged into a single, larger cycle. At the completion of Step 5, the minimal number of cycles will then have been obtained. Then again this set can be routed using only one additional wavelength, as described.

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 $\lceil P_{tot}/7\rceil$ 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 $\lceil P/7+1\rceil$ 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 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 non-blocking manner, nodes 1 and 2 must both have 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 *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 *one-to-two* exchange.

We first define some terminology which 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 counter-clockwise). 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 that 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 Figures 8 and 9. Finally, we have a general theorem for shifting converters if no adjacent converter

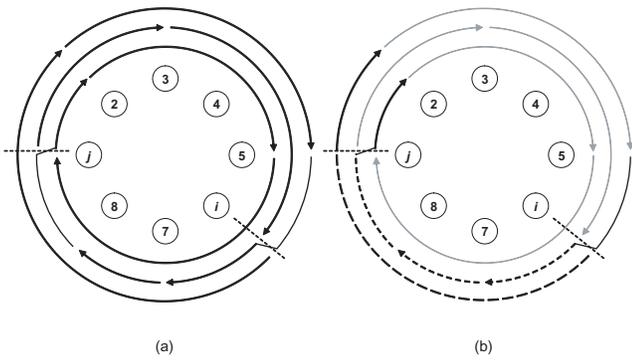


Fig. 8. (a) The original RWA of calls on the clockwise direction. Note that there is no requirement that the traffic set obey a P -port condition. Converters are used at nodes i and j . (b) The same ring, with related calls marked. Calls affected by the converter shifting are in bold, while unaffected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

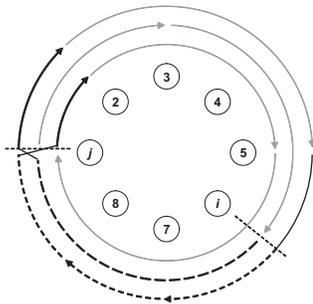


Fig. 9. All calls or parts of calls in the short dotted lines have exchanged wavelengths with those on the long dotted lines. Note that while a converter is no longer required at node i , an extra one is now being used at node j .

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 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 Figure 10. The proofs of the preceding theorems provide an algorithm for shifting converters from node to node. In the following two parts, 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.

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

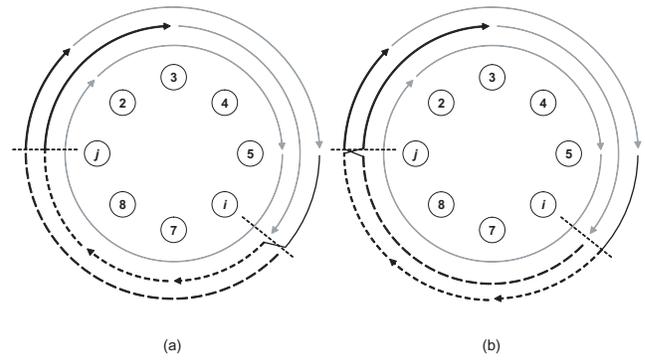


Fig. 10. (a) The original RWA of calls on the clockwise direction. A single converter is used by node i . Calls affected by the converter shifting are in bold, while unaffected calls are in light grey. The swap set consists of the dotted calls and parts of calls. (b) All calls or parts of calls in the 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 .

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 $i = 1, \dots, \lceil PN/4 \rceil - 2$.

Suppose node h is chosen to be the hub node. According to Lemma 6, we can move c_1 to node h using a one-to-two exchange. Next, move converter c_2 to node 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 $\lceil P/4 \rceil$ 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. (Note that the original algorithm required up to P converters at any given node.)

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 $2\lceil PN/7 \rceil$ converters.

2) *Symmetric Node Architecture*: It can be shown that after redistribution no more than $\lceil P/4 \rceil + 1$ converters are required at each node.

V. CONCLUSIONS

We considered the problem of implementing all virtual topologies on an N -node P -port network in a rearrangeably non-blocking fashion while trying to minimize the number of wavelengths and converters required. We show that 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 $\lceil PN/2 \rceil$ 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 $\lceil PN/7 \rceil$. 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.

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. 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.

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 [19], [20]. In our terminology, all-to-all traffic be-

longs 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 worst-case 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.

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, pp. 75–88, January 2002.
- [2] R. Ramaswami and K. N. Sivarajan, *Optical Networks: A Practical Perspective*, 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, pp. 1171–1182, July 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, pp. 598–607, October 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, pp. 186–198, April 2001.
- [6] R. Ramaswami and K. N. Sivarajan, "Routing and wavelength assignment in all-optical networks," *IEEE/ACM Trans. Networking*, vol. 3, pp. 489–500, October 1995.
- [7] A. Birman, "Computing approximate blocking probabilities for a class of all-optical networks," *IEEE JSAC*, vol. 14, no. 5, pp. 852–857, June 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, pp. 858–867, June 1996.
- [9] M. Kovacevic and A. Acampora, "Benefits of wavelength translation in all-optical clear-channel networks," *IEEE J. Select. Areas Commun.*, vol. 14, pp. 868–880, June 1996.
- [10] S. Subramaniam, M. Azizoglu, and A. Somani, "A new analytical model for multifiber WDM networks," *IEEE J. Select. Areas Commun.*, vol. 18, pp. 2138–2145, October 2000.
- [11] Y. Zhu, 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, pp. 747–762, December 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, pp. 89–96, January 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, pp. 186–196, April 1998.
- [14] L. Li and A. K. Somani, "Dynamic wavelength routing using congestion and neighborhood information," *IEEE/ACM Trans. Networking*, vol. 7, pp. 779–786, October 1999.
- [15] O. Gerstel, G. Sasaki, S. Kuttan, and R. Ramaswami, "Worst-case analysis of dynamic wavelength allocation in optical networks," *IEEE/ACM Trans. Networking*, vol. 7, pp. 833–845, December 1999.
- [16] R. Ramaswami and G. Sasaki, "Multiwavelength optical networks with limited wavelength conversion," *IEEE/ACM Trans. Networking*, vol. 6, pp. 744–754, December 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," in *JCIS. Joint Conference on Information Sciences*, March 2002.
- [18] M. N. S. Swamy and K. Thulasiraman, *Graphs, Networks, and Algorithms*, Wiley, New York, 1981.
- [19] 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, pp. 48–57, January 1999.
- [20] A. F. Elrafaie, "Multiwavelength survivable ring network architectures," in *Proc. ICC '93*, May 1993, pp. 1245–1251.