Dynamic Routing and Wavelength Assignment with Optical Bypass using Ring Embeddings

Li-Wei Chen, Eytan Modiano

Abstract— **We consider routing and wavelength assignment in ring, torus, and tree topologies with the twin objectives of minimizing wavelength usage and maximizing optical bypass. The P-port traffic assumption is used, which allows each node to send and receive at most P calls. For rings we show that $\lceil PN/4 \rceil$ wavelengths are necessary and sufficient, and provide a 4-hub ring architecture that requires only half of these wavelengths to be locally processed. We extend this approach to a torus by embedding virtual rings within the topology. For an $R \times C$ torus, we embed $R + C$ rings onto the torus and provide an approach to RWA and banding based on solving disjoint RWA/banding problems for each ring. Our RWA algorithm is more wavelength-efficient than any currently known algorithm and uses the minimum number of wavelengths for $R \geq 2C$. Finally, we give a RWA for trees that embeds a single virtual ring and uses the ring to obtain a RWA that requires no more than $\lceil PN/2 \rceil$ total wavelengths; this figure is shown to be optimal for balanced binary trees. Banding can be used for both the torus and the tree to further allow half the wavelengths to bypass all non-hub nodes.**

I. Introduction and System Model

WAVELENGTH division multiplexing (WDM) allows many connections to share the same fiber resources. The *routing and wavelength assignment* (RWA) problem becomes one of assigning routes and wavelengths to each call such that no two calls share the same wavelength on the same link. 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 the well-known 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. One way to categorize the existing literature is according to the traffic assumptions made. Two broad categories for traffic models are *static traffic* and *dynamic traffic*. In the static traffic model, the set of calls that needs to be supported by the network is fixed and does not change [1], [2]. In the dynamic model, calls are allowed to arrive and depart over time, requiring the RWA to change with the traffic. A variety of different models can be adopted for these dynamics. One method is to assume a statistical characterization of call arrival rates and holding times and design algorithms to minimize the call blocking probability [3], [4]. However, due to the large state-space size of the problem, the blocking probability of a WDM network is extremely difficult to analyze. An alternative approach, the *non-blocking model*, considers designing the network to accommodate any traffic matrix from an admissible set [5], [6]. Call arrivals or departures are equivalent to transitions 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*; otherwise, it is *rearrangeably non-blocking*. We adopt in this paper the P-port model used by [7], which defines the admissible set to be any traffic set where each node sends and receives at most P calls. 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 P-port traffic for ring and torus networks with and without wavelength conversion. We consider two approaches to designing RWA algorithms: minimizing the number of wavelengths in the network, and maximizing the amount of bypass at each node using banding.

The approach of minimizing the total number of wavelengths is common in the literature for nonblocking networks. Minimizing wavelengths is sensible for two reasons. First, adding additional wavelengths to a network is costly, so wavelengths should be used efficiently. Second, if each wavelength needs to be switched at each node, then optimizing the number of wavelengths is also equivalent to optimizing the number of switch ports at each node, an important property since switching is expensive and switching costs can rapidly dominate in the network as switch size increases.

The second approach is to use *banding* to reduce the number of wavelengths dropped at each node. Banding refers to the grouping of wavelengths into frequency bands; each band contains multiple adjacent wavelengths. Node complexity can be significantly reduced by allowing some bands to completely bypass each node. This is permissible if the RWA algorithm can guarantee that wavelengths within the bypassing bands are never dropped at those nodes.

For the purposes of the RWA problem, we can group the wavelengths into two bands: a *local* band, consisting of wavelengths that can be accessed by all nodes, and a *bypass* band, consisting of wavelengths that can be accessed only by a few designated *hub nodes*. The bypass band can therefore bypass the majority of the nodes in the network. There are several advantages to a banding approach. One is cost savings. Figure 1 shows an optical add-drop multiplexer (OADM) in a system where the total number of wavelengths W are divided into a local band of k wavelengths and a bypass band of $W - k$ wave-

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.

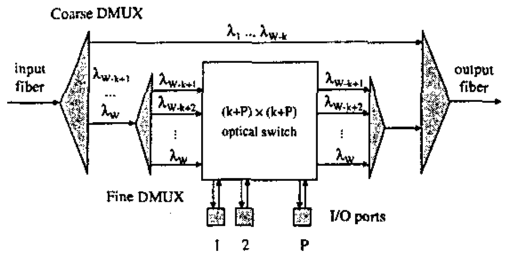

Fig. 1. An OADM architecture with bypass. Note that the switch size depends only on the number of wavelengths k entering the node, not the total number W.

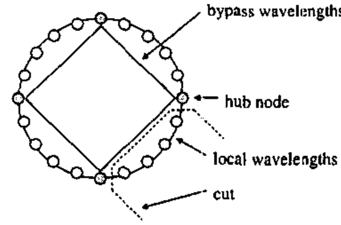

Fig. 2. A dual ring topology, equivalent to a 20-node, 4-hub ring with some local and some bypass wavelengths. The shaded nodes are hubs.

lengths. With banding, only the smaller local band of k wavelengths is switched. Without banding, the switch would have to process all W wavelengths. Another benefit is that the wavelength demultiplexers can be simpler: the first, coarse DMUX need only separate out two large bands, while the second, finer DMUX has a smaller band to work with (only the local wavelengths). Finally, by allowing wavelengths in the bypass band to avoid processing at non-hub nodes altogether, the bypass band can either avoid the switch (thus not suffering power losses due to switching which would reduce the reach of the lightpaths), or be placed in a separate fiber entirely. Such a separate fiber would need to be connected only to the hub nodes and could physically bypass all other nodes entirely.

We use the P-port traffic model, which restricts each node to sending and receiving no more than P calls. We provide novel RWA algorithms with and without bypass for ring and torus topologies. We demonstrate for each case optimal or efficient wavelength usage through comparisons with theoretical lower bounds, and demonstrate ways for using banding to further simplify node complexity. In Section II we cover RWA and banding algorithms for ring networks; Sections III and IV extend these results to torus and tree topologies, respectively.

II. Ring Networks

A. Routing and Wavelength Assignment

Recall that under the P-port traffic model, any traffic set in which each node sends and receives no more than P calls is admissible; as such, sufficiently many wavelengths must be provisioned to support all calls within any given admissible traffic set. In [7] it was shown that for a ring with $N > 7$ nodes, there exists an admissible traffic set which requires at least $\lceil PN/3 \rceil$ wavelengths to support it if no wavelength conversion is available. A RWA algorithm was also described that always uses no more than this minimum number of wavelengths for any admissible traffic set.

For rings with conversion, a cut-set bound can be used to provide a lower bound on the number of wavelengths required. Consider a cut of two links which divides the ring in equal halves of $N/2$ nodes each (N even). A worst-case admissible P-port traffic set can be constructed where each node on the left half of the cut sends all P units of traffic to some node on the right half. This means that $PN/2$ units of traffic would traverse 2 links, requiring a minimum of $\lceil PN/4 \rceil$ wavelengths. An algorithm was provided in [8] called the $\lceil PN/4 \rceil$ algorithm which achieves this lower bound, showing that the bound is sufficient as well. The algorithm requires a total of up to $\lceil PN/2 \rceil$ wavelength converters, which can be arbitrarily located within the ring. The details of the algorithm are omitted here; for the purposes of our discussion here it suffices to know that such an algorithm exists. This algorithm will be used later in the paper to assist in performing RWA on other topologies.

We point out that a corollary to the results of [8] is that for the special case of $N = 4$ and $P = 1$, only a single wavelength is required, and obviously no wavelength converters are required. This can be extended to the case of arbitrary P by noting that any P-port set can be decomposed into P single-port sets, and each set routed individually. Hence for $N = 4$ the lower bound of P wavelengths can be achieved without conversion.

B. Ring Banding Bound

In this section, we consider maximizing the amount of bypass using banding. Recall that the wavelengths are partitioned into two sets of adjacent wavelengths known as the local band and the bypass band. Wavelengths in the local band, known as local wavelengths, can be accessed by any node; wavelengths in the bypass band, known as bypass wavelengths, can be accessed only by a few special nodes known as hub nodes. Any non-hub node is termed a local node. As described in Section I, maximizing the size of the bypass band corresponds to reducing the overall cost of the network.

In general, the number of local wavelengths required decreases as the number of hubs increases. We next derive a lower bound on the size of the local band for a fixed number of hubs h within an N-node ring using a cut-set approach. The hub nodes divide the ring into sections, each consisting of a number of local nodes located between two consecutive hub nodes. Suppose the smallest such section consists of N_S local nodes. A traffic set exists where each node within that section sends all P calls to nodes outside that section. Imagine a cut consisting of the two links at the edge of that section. There are $P \cdot N_S$ units of traffic travelling across two links, and so a minimum of $\lceil P \cdot N_S/2 \rceil$ wavelengths are required. Furthermore, these must all be local wavelengths, since none of the local nodes can access bypass wavelengths. To obtain the tightest such bound, we maximize N_S by distributing the hubs symmetrically within the ring, resulting in $N_S = \lceil N/h \rceil - 1$ and a lower bound of $\lceil P(\lceil N/h \rceil - 1)/2 \rceil$. This situation is illustrated in Figure 2 for the case of $h = 4$.

C. Banding and Bypass on Rings

In this section, an algorithm called the *ring banding algorithm* that minimizes the number of wavelengths in the local band is described. In general, there exists an entire class of ring banding algorithms depending on the number of hubs in the ring; however, from a single instance of the algorithm it is straightforward to extrapolate to any other variation since the algorithms are all very similar. Therefore in this paper we focus primarily on the special case of a 4-hub ring banding algorithm. A 4-hub architecture has the added advantage that routing traffic between the four hubs does not require wavelength conversion to achieve optimal wavelength efficiency, as noted in the preceding section, since the hubs form a 4-node ring.

For $h = 4$ hubs, the lower bound is $\lceil P(\lceil N/h \rceil - 1)/2 \rceil = \lceil P(\lceil N/4 \rceil - 1)/2 \rceil$ wavelengths. To obtain some intuition about how large this local band is, consider the case where N is a multiple of 4; under this assumption, this minimum number becomes $\lceil \frac{P(N-4)}{8} \rceil$. This establishes that at least half of all the wavelengths must be local wavelengths. (Recall that $\lceil PN/4 \rceil$ total wavelengths are required.)

To reduce the number of local wavelengths required, consider a topology where the 4 hubs are distributed symmetrically within the ring. The ring banding algorithm gives each call a route and wavelength assignment using a three-step process, as follows:

1) Starting from the source node, the call travels to the nearest hub. This route uses a local wavelength and is static since the nearest hub node for any given source node is fixed.
2) From that hub, it travels via a bypass wavelength to the hub closest to the destination node. This routing is dynamic since the source and destination of the call are variable.
3) Finally, the call proceeds from that hub to the destination node via a local wavelength, again by a static route.

We first prove that Steps 1 and 3 use no more than the minimum number of local wavelengths. Then we provide an RWA to dynamically route all the calls in Step 2 using as few bypass wavelengths as possible.

Consider the local wavelength usage. The hubs partition local nodes in the ring into four quarters, each of which contains no more than $\lceil N/4 \rceil - 1$ local nodes. Assign each node within the quarter P bidirectional wavelengths for communication with the hub closest to it. The forward direction, from the node to the hub, is used in Step 1; the reverse direction, from the hub to the node, is used in Step 3. Since half the nodes in each quarter communicate with the hub node on one side and half with the other, each bidirectional wavelength can be shared by two local nodes. This is illustrated for a 16-node ring in Figure 3. In total, $\lceil P(\lceil N/4 \rceil - 1)/2 \rceil$ local wavelengths are required by this scheme. Note that this meets the lower bound on local wavelengths, and corresponds to roughly half of the total number of wavelengths used by the $\lceil PN/4 \rceil$ algorithm.

Next we consider the dynamic routing of calls between hubs in Step 2. Each hub is responsible for sending and receiving all calls belonging to the $\lceil N/4 \rceil$ nodes closest to it (including itself) to and from other hubs. Therefore, for Step 2 each of the 4 hubs acts as though it has $P' = P\lceil N/4 \rceil$ ports. From the results

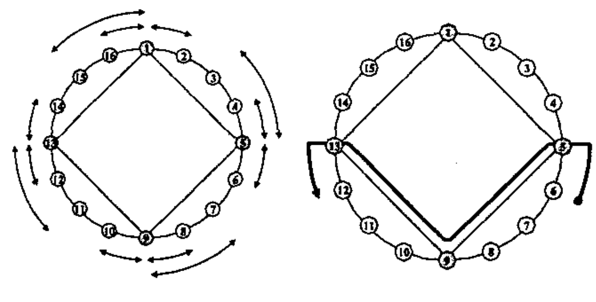

Fig. 3. (a) Assigning local wavelengths for a single-port 16-node ring. (b) The route for call (6, 12). Hubs 5 and 13 are used to access the bypass wavelengths.

of Section II-A, we know that $P\lceil N/4 \rceil$ bypass wavelengths are necessary and sufficient to support such traffic, and furthermore that we can do so without the use of wavelength converters. (To see this, note that in the 4-hub ring $N = 4$ and $P' = P\lceil N/4 \rceil$.)

Local Node Requirements: Each local node can use fixed routes and wavelengths to communicate with its nearest hub node. This allows the local node architecture to be extremely simple. Not only do local nodes never need to access bypass wavelengths, they do not need to switch local wavelengths either since these wavelengths are statically assigned. Therefore, the switching block shown in Figure 1 can be replaced with static connections from the P ports to the P assigned wavelengths.

Hub Node Requirements: Each of the four hub nodes needs to be able to switch between any of the local wavelengths and any of the bypass wavelengths. Bypass wavelengths never need to be switched onto other bypass wavelengths – i.e. no conversion is required between bypass wavelengths. Conversion is required between bypass and local wavelengths.

We illustrate the operation of the 4-hub ring banding algorithm with an example.

Example 1: Figure 3 considers single-port traffic on a 16-node ring, and shows the wavelengths that would be assigned in Steps 1 and 3 of the algorithm to handle the traffic between the local nodes and the hubs. Note that only the minimum number $\lceil (16/4 - 1)/2 \rceil = 2$ of local wavelengths are required. For any fixed traffic set, the four hubs would then provide a RWA using the bypass wavelengths via the $\lceil PN/4 \rceil$ algorithm.

To illustrate the full RWA for a given call, consider a call travelling from node 6 to node 12. The closest hub for node 6 is 5, and the closest hub for node 12 is 13. Therefore a route is assigned as follows: (1) from node 6 to node 5 via a local wavelength, (2) from node 5 to node 13 via a bypass wavelength, and (3) from node 13 to node 12 via a local wavelength again. Figure 3 shows an example of a possible RWA for the call using this approach.

III. Torus Networks

A. Torus RWA Lower Bound

Torus networks with no wavelength conversion were considered in [9], which presented a suboptimal RWA algorithm based on column-first routing requiring twice the minimum number of wavelengths. We investigate the use of wavelength conversion to reduce the number of wavelengths required. We will first show a lower bound on wavelengths required; subsequently, we give a novel algorithm for RWA on a torus that makes efficient use of wavelengths and achieves the bound in certain cases.

Define an $R \times C$ torus to be a network consisting of RC nodes, each of which is uniquely assigned to a row r, $1 \leq r \leq R$, and a column c, $1 \leq c \leq C$. Each node is connected via four bidirectional links to four other nodes: the two adjacent nodes occupying the same row, and the two adjacent nodes in the same column. More specifically, denote the node in row r and column c by $n_{r,c}$. Then the node $n_{r,c}$ is connected via bidirectional links to the nodes $n_{r\oplus 1,c}$, $n_{r\ominus 1,c}$, $n_{r,c\oplus 1}$, and $n_{r,c\ominus 1}$, where operations on the rows and columns are modulo R and C, respectively. Figure 4 shows an example of a 6×3 torus. **Note that each row contains C nodes, and each column contains R nodes.**

We again use the P-port traffic model, where each node is assumed to have P ports, and hence can send and receive at most P calls. We next investigate how many wavelengths must be provisioned in the torus to support any admissible traffic set under these assumptions.

Suppose without loss of generality that $R \geq C$. (If the opposite is true, rotate the picture of the torus 90 degrees and re-label the rows as the columns and vice versa.) Let R be even. Consider a horizontal cut across the columns which removes $2C$ links and divides the torus into two equal sets of $RC/2$ nodes each. Consider the first set. Under the P-port model, there exists a worst-case traffic set in which each node in that set sends all P calls to some node in the other set. In this case, there are $PRC/2$ calls traversing $2C$ links, which means that a minimum of $\lceil (PRC/2)/(2C) \rceil = \lceil PR/4 \rceil$ wavelengths are required. A similar argument for R odd yields a bound of $\lceil P(R-1)/4 \rceil$ wavelengths.

B. The TERA Algorithm

In this section we describe an algorithm based on the Torus Embedded-Ring Approach (TERA) for routing and wavelength assignment. We will show that by judicious use of wavelength conversion, the TERA algorithm will use no more than $\max\{\lceil PC/2 \rceil, \lceil PR/4 \rceil\}$ wavelengths. For toruses where $R \geq 2C$, this achieves the lower bound of $\lceil PR/4 \rceil$ wavelengths; in the worst case ($R = C$) it uses $\lceil PR/2 \rceil$.

We will describe the algorithm in detail later in this section, but the general idea is as follows. For any given call going from n_{r_1,c_1} to n_{r_2,c_2}, instead of considering all possible route assignments (of which there are many), we break the problem down into finding a route from the source n_{r_1,c_1} to some intermediate node n_{r_1,c_b} in the same row, from n_{r_1,c_b} to some n_{r_2,c_b} in the same column, and finally from n_{r_2,c_b} to the destination n_{r_2,c_2} in the same row. The advantage to this approach is that instead of having a single call travelling through a torus, the call has been subdivided into three smaller calls, each on a different ring, and the results for rings in the preceding section can be used to do the routing and wavelength assignment for each sub-call. Figure 4 gives an example of routing a call using this approach.

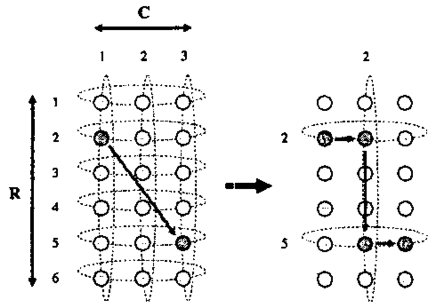

Fig. 4. Breaking up a call into three sub-calls using a bridging column. The single call on the left, from $n_{2,1}$ to $n_{5,3}$, is routed as two row-ring calls and a column-ring call using the bridging column 2. Each of the sub-calls can be routed independently on their respective rings using the $\lceil PN/4 \rceil$ algorithm. Additional wavelength conversion may be required at nodes $n_{2,2}$ and $n_{5,2}$ if the sub-calls are not assigned to the same wavelength.

In the above approach, an intelligent choice of a column c_b for each call is required so that subsequent routing of the resulting sub-calls then uses as few wavelengths as possible.

For notational purposes, define the ring formed by the nodes $\{n_{i,j} | j = 1, ..., C\}$ to be the row-ring i, and the ring formed by the nodes $\{n_{i,j} | i = 1, ..., R\}$ to be the column-ring j. Under this nomenclature, there are R row-rings and C column-rings. Let us call the columns $\{c_b\}$ used to generate the sub-calls the *bridging columns*, since calls will use these columns to travel between row-rings. In an $R \times C$ torus, there are C bridging columns. We will use a bipartite matching approach to associate each call with a bridging column in such a way that the resulting sub-calls will form a $2P$-port traffic set on each row-ring, and a P-port traffic set on each column-ring. Once this is done, it is evident that by using the $\lceil PN/4 \rceil$-algorithm on each row and column-ring, the total number of wavelengths required in the torus will be $\max\{\lceil 2PC/4 \rceil, \lceil PR/4 \rceil\}$ $= \max\{\lceil PC/2 \rceil, \lceil PR/4 \rceil\}$, as claimed.

C. Bridging Column Assignment

In this section, we will describe a method for using matchings to assign a bridging column to each call such that the resulting sub-calls form a $2P$-port traffic set on each row-ring, and a P-port traffic set on each column-ring.

Consider a call $(n_{r_1,c_1}, n_{r_2,c_2})$ that is divided into three sub-calls $(n_{r_1,c_1}, n_{r_1,c_b})$, $(n_{r_1,c_b}, n_{r_2,c_b})$, and $(n_{r_2,c_b}, n_{r_2,c_2})$. We will call the $(n_{r_1,c_1}, n_{r_1,c_b})$ the *starting sub-call*, $(n_{r_1,c_b}, n_{r_2,c_b})$ the *bridging sub-call*, and $(n_{r_2,c_b}, n_{r_2,c_2})$ the *ending sub-call*. We wish to determine the conditions that the the bridging column assignment are subject to.

1. Row-ring conditions: For each row-ring of size C, there are $2PC$ sub-calls to be routed on it, P of which are starting sub-calls and the remaining P of which are ending sub-calls. Each node already uses P ports for sending the starting sub-calls, and P ports for receiving the ending sub-calls. We wish to choose bridging columns such that the starting sub-calls on each row-ring use no more than P additional destination ports per node, and the ending sub-calls use no more than P additional source ports per node. If these conditions are met, then each row-ring needs no more than $2P$ ports per node.

2. Column-ring conditions: For each column-ring, we wish to choose bridging columns such that the P-port assumption holds true for each column-ring; i.e. the set of PR bridging calls for each column-ring should use no more than P source and destination ports per node.

Lemma 1: If the bridging conditions imposed by the column-rings are satisfied, then the bridging conditions for the row-rings are also satisfied.

Proof: For each row-ring, at least P ports are required regardless of the choice of column-rings since the sources of each starting sub-call and the destinations of each ending sub-call are fixed. Thus it remains to show only that each node is the destination of no more than P starting sub-calls, and the source of no more than P ending sub-calls, to prove that only another P ports per node are required (for a total of $2P$ per node).

To see that this is true, note that the destination of each starting sub-call is the source of a corresponding bridging sub-call. Therefore, if a node in a row-ring were the destination of more than P starting sub-calls, that node would also be the source of more than P bridging sub-calls on its column-ring. Since we assumed that the column-ring conditions were satisfied, this cannot be true.

Similarly, the source of each ending sub-call is the destination of a corresponding bridging sub-call. If a node in a row-ring were the source of more than P ending sub-calls, that node would also be the destination of more than P bridging sub-calls on its column-ring, which cannot be true. ∎

Lemma 1 tells us that it suffices to assign calls to bridging columns so that the column-ring conditions are satisfied. We achieve this by using a bipartite matching approach, as follows.

Consider the traffic set for the $R \times C$ torus. We construct a bipartite graph consisting of two sets of R vertices, which we will call $\{S_i\}$ and $\{D_j\}$, where $i = 1, \ldots, R$ and $j = 1, \ldots, R$. In a given set, each vertex corresponds uniquely to one of the R row-rings.

Each call in the traffic set will correspond to an edge in the graph. A call from a node n_{r_1,c_1} to n_{r_2,c_2} will be represented by an edge from vertex S_{r_1} to D_{r_2}; in other words, a call in the torus is represented by an edge in the graph connecting the vertex in $\{S_i\}$ representing its source row-ring, and the vertex in $\{D_j\}$ representing its destination row-ring. Since we have a P-port traffic set, and each row-ring contains C nodes, each vertex in the graph will have degree PC.

We will call the bipartite graph thus constructed the *bridging graph.* In the next step, we will make use of the following theorem for bipartite matchings with equal nodal degree.

Theorem 1: Define a *perfect matching* to be a set of edges where exactly one edge is incident on every vertex. Then, in a bipartite graph $(V_1, V_2, \mathcal{E})$ in which each vertex in V_1 and in V_2 has degree m, the set $\mathcal{E}$ can be partitioned into m disjoint perfect matchings.

Proof: The proof is basically by induction using Hall's theorem and is omitted here for brevity. ∎

In the context of our constructed bipartite matching, Theorem 1 guarantees that we can obtain a set of PC disjoint perfect matchings. Each perfect matching corresponds to a set of R calls where exactly one call originates from each row-ring, and one call is destined for each row-ring. Since for any given call

source		destination		graph edge	
row	column	row	column	S_i	D_j
1	1	3	2	1	3
1	2	2	1	1	2
2	1	1	1	2	1
2	2	4	1	2	4
3	1	2	2	3	2
3	2	1	2	3	1
4	1	4	2	4	4
4	2	3	1	4	3

(a)

call		bridging column
src row	src col	
1	1	1
1	2	2
2	1	1
2	2	2
3	1	1
3	2	2
4	1	1
4	2	2

(b)

Fig. 5. (a) A traffic set for the single-port 4 × 2 torus considered in Example 2. The first two pairs of columns give the row-column pairs for the source and destination nodes, while the last two columns give the edges that represent each respective call in the bridging graph. (b) The resultant assignment of bridging columns to calls for Example 2

the bridging sub-call has source node equal to the source row of the original call, and destination node equal to the destination row of the original call, this means that if all calls in a matching use the same bridging column, *the set of resultant bridging sub-calls will correspond to a single-port traffic matrix for that column.*

The preceding idea forms the basis for the assignment of bridging columns. Recall that Theorem 1 guarantees that we will have PC disjoint perfect matchings. Divide these matchings into C sets of P disjoint perfect matchings. Assign each set of matchings to one of the C columns. All calls in a matching assigned to a given column will use that column as its bridging column. Since each matching requires only a single port per node, the P matchings in each column will require no more than P ports per node. Thus the column-ring conditions (and subsequently the row-ring conditions, by Lemma 1) are satisfied.

Example 2: In this example, we consider the problem of assigning bridging columns to a traffic set on a single-port 4 × 2 torus. The traffic set is given in Figure 5. The corresponding bridging graph is shown in Figure 6. Recall that Theorem 1 states that since the graph has vertex degree $PC = 1 \cdot 2 = 2$, we can find 2 disjoint matchings. One possible such choice is given.

Under the choice of matchings given in Figure 6, we assign all calls in the first matching to column 1, and all calls in the second matching to column 2. This uniquely specifies a bridging column for each call in the traffic set, as shown in Figure 5. For example, the first call in the set, from $n_{1,1}$ to $n_{3,2}$, corresponds to the graph edge S_1 to D_3. This edge is in the first matching, so the call from $n_{1,1}$ to $n_{3,2}$ is assigned to the bridging column of 1. The resulting sub-calls are $(n_{1,1}, n_{1,1})$, $(n_{1,1}, n_{3,1})$, and $(n_{3,1}, n_{3,2})$. (Note that one of the sub-calls happened to be degenerate.)

We can now formally state the TERA algorithm.

THE TERA ALGORITHM

1) Given a P-port traffic set for the torus, construct the corresponding bipartite bridging graph, as described in Section III-C. Divide the edges on the bridging graph into C sets of P disjoint bipartite matchings. Assign each set to a different bridging column.
2) Now that each call has a bridging column, divide each call into a starting sub-call, a bridging sub-call, and an

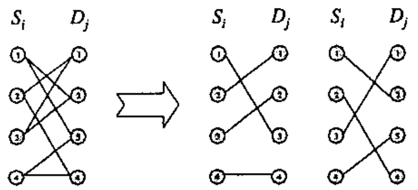

Fig. 6. The bridging graph for Example 2. As expected, each vertex in the bridging graph has vertex degree $PC = 1 \cdot 2 = 2$. Using Theorem 1, this can be divided into $C = 2$ disjoint perfect matchings.

ending sub-call.

3) For each row-ring, use the $\lceil PN/4 \rceil$ algorithm to perform RWA on all starting and ending sub-calls within that row-ring. This requires $\lceil PC/2 \rceil$ wavelengths per row-ring.
4) For each column-ring, use the $\lceil PN/4 \rceil$ algorithm to perform RWA of all bridging sub-calls within that column-ring. This requires $\lceil PR/4 \rceil$ wavelengths per column-ring.
5) Give each original call in the torus the route and wavelength assignment formed by the concatenation of the RWA of the starting, bridging, and ending sub-calls. Up to two converters may be needed to change between sub-calls.

D. Banding and Bypass on Toruses

In this section, we give an algorithm for banding on a torus which reduces size of the band of local wavelengths dropped at non-hub nodes.

Since the TERA algorithm essentially results in a problem of disjointly routing traffic on different rings in the rows and columns, we can apply an approach similar to the ring banding algorithm. Rather than using the $\lceil PN/4 \rceil$ algorithm to route the sub-calls on the row-rings and column-rings, the ring banding algorithm is used. In this discussion, we focus on using the 4-hub ring banding algorithm, but the results could be extended to using different numbers of hubs.

We assume again by convention that $R \geq C$. Using a 4-hub architecture on each row, we can reduce the number of local wavelengths required on those rings to $P\lceil C/8 \rceil$. A minimum of $4R$ hubs are required to do this, since no two row-rings can share a hub node. In order for rings along the columns to also require a local band of no more than $P\lceil C/8 \rceil$ wavelengths, enough hubs must be allocated along the columns such that no node in a column-ring is further than $\lceil C/8 \rceil$ hops from a hub node. This requires at least $\lceil \frac{R}{\lceil C/4 \rceil} \rceil$ hub nodes along each column. An upper bound on the total number of hubs required is therefore $4R + \lceil \frac{R}{\lceil C/4 \rceil} \rceil \cdot C$, or approximately $8R$ hubs. Therefore the number of hubs h^* is bounded by $4R \leq h^* \leq 8R$. We can achieve the lower bound by using clever hub designation to allow the same nodes to serve as hubs for both a row-ring and a column-ring, reducing the total number of hubs required.

We describe a hub allocation scheme that uses the minimum number of hubs. For the first row, designate nodes $n_{1,h_1}, n_{1,h_2}, n_{1,h_3}$, and n_{1,h_4} to be hubs, where the column numbers h_i are given in Table I. In the second row, nodes $n_{2,1\oplus h_1}, n_{2,1\oplus h_2}, n_{2,1\oplus h_3}$, and $n_{2,1\oplus h_4}$ are hubs; note that this

TABLE I

HUB COLUMN NUMBERS FOR THE FIRST ROW-RING

If C mod 4 = ...	h_1	h_2	h_3	h_4
0	0	$C/4$	$2\,C/4$	$3\,C/4$
1	0	$\lfloor C/4 \rfloor$	$2\lfloor C/4 \rfloor$	$2\lfloor C/4 \rfloor + \lceil C/4 \rceil$
2	0	$\lceil C/4 \rceil$	$\lfloor C/4 \rfloor + \lceil C/4 \rceil$	$\lfloor C/4 \rfloor + 2\lceil C/4 \rceil$
3	0	$\lceil C/4 \rceil$	$2\lceil C/4 \rceil$	$3\lceil C/4 \rceil$

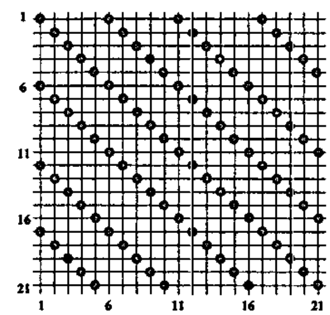

Fig. 7. A 21 × 21 torus. The local nodes are at all intersection points of the grid, while the hub nodes are shown as shaded circles. For $R = C = 9$, we have that $\lfloor N/4 \rfloor = 2$ and $\lceil N/4 \rceil = 3$, so that $h_1 = 0$, $h_2 = 2$, $h_3 = 4$, and $h_4 = 7$. Adding each of the row numbers modulo 9 gives the hub assignments shown. As a check, note that $g_1 = 0$, $g_2 = 2$, $g_3 = 5$, and $g_4 = 7$ also correctly yields the resulting hub allocations down the columns.

is a cyclic shift in the right-ward direction (modulo C) of the hub allocation for the previous row. This pattern repeats for all subsequent rows. In general, $n_{r,c}$ is a hub if $c \oplus (r-1) = h_i$ for some $i = 1, 2, 3$, or 4; i is called the *hub index* of that hub.

It is trivial to note that each row now has 4 hubs, and hence only $P\lceil C/8 \rceil$ local wavelengths are required along the rows. The following lemma claims that this hub allocation also requires no more than $P\lceil C/8 \rceil$ local wavelengths along the columns.

Lemma 2: Designating hub nodes along the row-rings as described results in a hub allocation with the property that along each column-ring, no local node is more than $\lceil C/8 \rceil$ hops away from the nearest hub node.

Proof: The proof will show that along any column, no two hubs are separated by more than $\lceil C/4 \rceil$ nodes, from which it follows that no node can be more than $\lceil C/8 \rceil$ hops away from a hub.

Recall that a node $n_{r,c}$ is a hub if $c \oplus (r-1) = h_i$; equivalently, $n_{r,c}$ is a hub iff $r = h_i \ominus (c-1)$ for some $i = 1, 2, 3$, or 4. Consider two adjacent hubs in the same column $n_{r_1,c}, n_{r_2,c}$ with hub indices i and j; the distance between them is $r_1 - r_2 = h_i - h_j$. From Table I, for any two adjacent h_i and h_j, $h_i - h_j \leq \lceil C/4 \rceil$, completing the proof. ■

The consequence of the lemma is that any local node is no more than $\lceil C/8 \rceil$ nodes away from a hub; hence no more than $P\lceil C/8 \rceil$ wavelengths are required. Figure 7 gives an example of a 21 × 21 torus, and illustrates the hub allocation obtained from this construction.

The number of bypass wavelengths required can be obtained by examining the number of calls arriving and departing from each hub node. Along the rows, each hub is responsible for $\lceil C/4 \rceil$ nodes, and each such node has $2P$ ports; therefore each row-hub has $P' = 2P\lceil C/4 \rceil$. Using the $\lceil PN/4 \rceil$ algorithm,

$P\lceil C/2 \rceil$ bypass wavelengths are required. A similar argument on the columns shows that $P\lceil R/4 \rceil$ bypass wavelengths are required.

The wavelength assignment also has two useful properties. First, non-hub nodes have fixed routing and fixed wavelengths. Conversion between local wavelengths is hence not required, since each node is assigned its own wavelength to send and receive from its hub. Also, local nodes need no knowledge of network state. Second, the only place where conversion is required is at the hubs. No conversion is required to connect calls continuing from a row-ring onto a column-ring, or from a column-ring onto a row-ring.

IV. Tree Networks

A. Tree RWA Lower Bound

In this section, a bound on the minimum number of wavelengths required to support P-port traffic in trees is established. We use the cut-set bound to obtain a lower bound on the number of wavelengths. Since a tree contains no cycles, the removal of any single link i disconnects the tree into two disjoint sets of nodes. For each i, we call these sets S_i^1 and S_i^2, and let $|S_i^1|$ and $|S_i^2|$ denote the number of nodes in each set, respectively. Suppose $|S_i^1| \leq |S_i^2|$. Then there exists a worst-case admissible set where each node in S_i^1 sends all P units of traffic to some node in S_i^2. Since all this traffic must cross link i, at least $W_i = |S_i^1|$ wavelengths are required to support it. If $|S_i^1| > |S_i^2|$, similar reasoning gives $W_i = |S_i^2|$. We can obtain the tightest lower bound by maximizing over all links i. Let the greatest lower bound thus obtained be $W = \max_i\{W_i\} = \max_i\{\min\{|S_i^1|, |S_i^2|\}\}$. A link which achieves this lower bound is known as a *bottleneck link.*

From the preceding discussion, it is clear that the lower bound obtained in this section is dependent on the specific topology of the tree, and not just on the number of nodes in the tree. For example, for a balanced d-ary tree, the links adjacent to the root node are the bottleneck links, and $W = P(N-1)/d$. For balanced binary trees, where $d = 2$, the bound is $W = P(N-1)/2$.

B. The $\lceil PN/2 \rceil$ Embedded-Ring Approach

[10] has considered RWA on trees using a bipartite matching approach. In this section, we describe a novel RWA based on embedding a virtual ring in the tree topology. We will show that for connected P-port traffic sets, this approach requires at most $\lceil PN/2 \rceil$ wavelengths for any tree topology, and hence is optimal for tree topologies where $W = \lceil PN/2 \rceil$. For example, it is optimal for balanced binary trees. Furthermore, no wavelength conversion is required.

The ring-embedding idea is intuitively very simple. In any tree, by using depth-first search, we can form a circuit which visits each node in the tree at least once while traversing each link only twice (once in each direction). This circuit is said to form a *virtual ring* in the following sense. Consider a ring topology where the nodes are connected in the order in which each corresponding node in the tree is first visited by the circuit. Then any RWA for this ring has a one-to-one correspondence with a RWA for the tree. Each link between two adjacent nodes on the ring corresponds to the links traversed by the circuit in travelling between those two nodes on the tree. Such a circuit is illustrated in Figure 8 for a 15-node balanced binary tree.

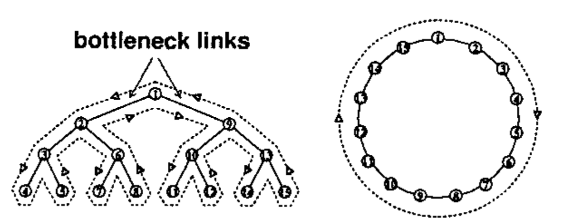

Fig. 8. Embedding a cycle in a 15-node balanced binary tree. The nodes have been numbered so that the cycle visits them in order of increasing index. The corresponding virtual ring topology is shown next to the tree.

A single, unidirectional wavelength on the ring corresponds to the use of a single, bidirectional wavelength on the tree. (A bidirectional wavelength is required because a single circuit around the tree used each link once in each direction.) Define two calls on a ring to be *adjacent* if the destination of one call is the source of the other. For the RWA on the ring, we rely on a lemma from [7] which states that any two adjacent calls on a ring must fit on a single wavelength in one direction or the other. (A graphical proof is straightforward.) The ring embedding algorithm simply divides the traffic into adjacent pairs, and determines the single-wavelength direction for each pair. Each pair is then routed on a single directed wavelength on the virtual ring, which corresponds to a RWA on the tree which uses a single bidirectional wavelength per pair. Since there are a total of $\lceil PN/2 \rceil$ pairs, no more than $\lceil PN/2 \rceil$ wavelengths are needed.

We can also extend the ring banding results to the tree by using the ring banding algorithm on the virtual ring. It can be shown that this is analogous to the ring and torus cases and allows half the wavelengths to bypass all but the hub nodes.

References

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

[2] 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.

[3] A. Birman, "Computing approximate blocking probabilities for a class of all-optical networks," *IEEE J. Select. Areas Commun.*, vol. 14, no. 5, pp. 852–857, June 1996.

[4] 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, pp. 747–762, December 2000.

[5] 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, pp. 833–845, December 1999.

[6] R. Ramaswami and G. Sasaki, "Multiwavelength optical networks with limited wavelength conversion," *IEEE/ACM Trans. Networking*, vol. 6, pp. 744–754, December 1998.

[7] 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.

[8] L.-W. Chen and E. Modiano, "Efficient routing and wavelength assignment for reconfigurable WDM networks with wavelength converters," in *Proc. IEEE INFOCOM*, April 2003.

[9] P. Saengudomlert, E. Modiano, and R. G. Gallager, "On-line routing and wavelength assignment for dynamic traffic in WDM ring and torus networks," in *Proc. IEEE INFOCOM*, April 2003.

[10] P. Saengudomlert, E. H. Modiano, and R. G. Gallager, "Dynamic wavelength assignment for wdm all-optical tree networks," in *Proc. Allerton*, September 2003.