Distributed Scheduling Algorithms for Wavelength Convertible WDM Optical Interconnects

Zhenghao Zhang and Yuanyuan Yang
Department of Electrical and Computer Engineering
State University of New York, Stony Brook, NY 11794, USA

Abstract— **Optical communication is attracting more and more attentions because of its huge bandwidth to meet the ever increasing demand of emerging computing/networking applications. In this paper we study distributed scheduling algorithms to resolve output contentions in WDM optical interconnects with wavelength conversion ability. We consider the general case of limited range wavelength conversion, including the full range wavelength conversion. Two types of limited range wavelength conversions, circular symmetrical and non circular symmetrical, are studied. We introduce the request graph and show that finding the largest group of contention-free connection requests to achieve maximum network throughput is equivalent to finding a maximum matching in the request graph. Compared with the existing algorithm for finding a maximum matching in an arbitrary bipartite graph with time complexity $O(N^{\frac{3}{2}}k^{\frac{3}{2}}d)$, the algorithms we present have time complexity of $O(k)$ and $O(dk)$ (independent of interconnect size N) for non-circular symmetrical and circular symmetrical wavelength conversion, respectively, where k is the number of wavelengths per fiber and d is the conversion degree. In addition, our algorithms can be easily implemented in hardware, and used for time slotted WDM optical interconnects where connections hold for different number of time slots.**

I. Introduction and Background

Many emerging computing/networking applications, such as data-browsing in the world wide web, video conferencing, video on demand, E-commerce and image distributing, require very high network bandwidth often far beyond that today's high-speed networks can offer. Optical networking is a promising solution to this problem because of the huge bandwidth of optics: a single fiber has a bandwidth of nearly 50 THz [16]. To fully utilize the bandwidth, a fiber is divided into a number of independent channels, with each channel on a different wavelength. This is referred to as *wavelength-division-multiplexing (WDM)*.

In a WDM all optical network, data is modulated on a selected wavelength channel and this information-bearing signal remains in the optical domain throughout the path from source to destination. In the absence of wavelength conversion ability, the signal is required to be on the same wavelength from hop to hop, which is referred to as the *wavelength continuity constraint*. This constraint can be removed when *wavelength converters* are employed in the network. Wavelength converter converts a signal on one wavelength to another wavelength, and makes the network more flexible for satisfying various connection requests. Studies show that network performance is greatly improved by using wavelength converters [6]. If a wavelength converter can convert a wavelength to any other wavelength in the optical system, it is called *full range wavelength converter*. However, this type of wavelength conversion is quite difficult and expensive to implement due to technological limitations [13], [11]. A realistic all-optical wavelength converter may only be able to convert to a limited number of wavelengths for any given wavelength, which is called *limited range wavelength converter*. In general, a limited range wavelength converter can convert a wavelength to a set of adjacent wavelengths, and the number of wavelengths in the set is called *conversion degree*. Research [13], [11], [14] shows that limited range wavelength converters can achieve network performance similar to full range wavelength converters even when the conversion degree is very small. Thus, limited range converters are considered as a practical, cost-effective choice for providing wavelength conversion ability in WDM networks, which will be the main focus of this paper.

Research supported in part by the U.S. National Science Foundation under grant numbers CCR-0073085 and CCR-0207999.

A *WDM optical interconnect* (also called WDM switch in some literature) provides interconnections between a group of input fiber links and a group of output fiber links with each fiber link carrying multiple wavelength channels. Such an optical interconnect can be used to serve as a crossconnect (OXC) in a wide-area communication network or to provide high-speed interconnections among a group of processors in a parallel and distributed computing system. The WDM interconnect we consider is an $N \times N$ switch, i.e., there are N fibers on the input side of the switch and N fibers on the output side of the switch. On each fiber there are k wavelengths that carry independent data. Thus, there are a total of Nk input wavelength channels and Nk output wavelength channels in the interconnect. Any input wavelength channel can be connected to any output fiber. In addition, there are limited range wavelength converters with conversion degree d ($d \leq k$) equipped on the output side of the switch. With limited range wavelength conversion, an input wavelength channel may be connected to d adjacent channels on an output fiber. Clearly, full range wavelength conversion is a special case of limited range conversion when $d = k$. Figure 1 shows an $N \times N$ WDM interconnect with k wavelengths on each fiber, in which each wavelength can be converted to d adjacent wavelengths. It can be seen from the figure that an input fiber is first fed into a demultiplexer, where different wavelength channels are separated one from another. The separated wavelength channels are the input of a switching fabric, through which an input wavelength channel is connected to d adjacent wavelength channels on an output fiber. At the output side of the switching fabric, there are optical combiners for each output wavelength channel which is used to combine

0-7695-1926-1/03/$17.00 (C) 2003 IEEE
Proceedings of the International Parallel and Distributed Processing Symposium (IPDPS'03)

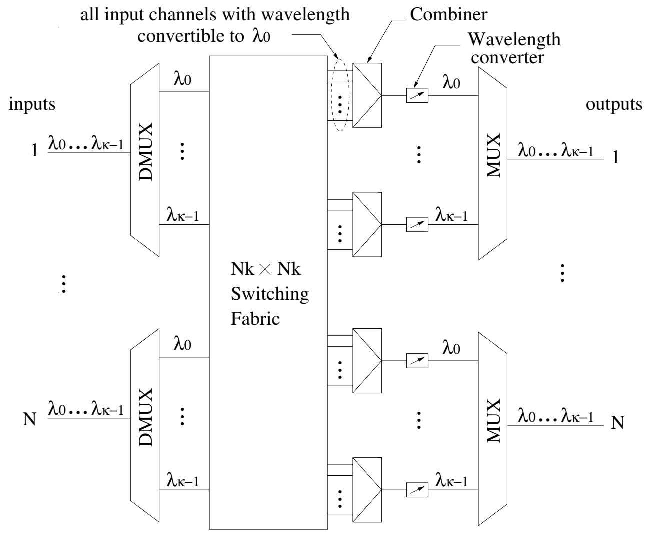

Fig. 1. A wavelength convertible WDM optical interconnect.

the optical signals that are able to access this output wavelength channel. Thus, there are Nd inputs to a combiner, but only one of them may carry signal at a time. The output of the combiner is then fed into a wavelength converter, which converts the wavelength of the signal to the desired wavelength. Then the k output wavelengths are multiplexed into an output fiber.

The WDM optical interconnect can be operated either asynchronously or synchronously. The former case applies to WDM wavelength routing networks that are similar to electrical circuit switching networks. A typical scenario for the latter case would be an optical WDM packet switching network where information is carried in optical packets that arrive to the interconnect at the beginning of each time slot. The duration of an optical packet is usually assumed to be one time slot. However, we also consider the general case of multiple time slot duration in this paper. All connections or optical packets are assumed to be of the same priority. Since optical buffers are currently made of fiber delay lines and are still very expensive [5], we also assume that there are no buffers in the WDM optical interconnect. The traffic pattern considered in this paper is unicast, i.e., each connection request is destined for only one output fiber. The connection request does not specify which wavelength channel on the destination fiber it should be connected to, and we can assign to it any free wavelength channel accessible to this connection. Under these assumptions, the connection requests arrived at the interconnect in one time slot can be partitioned into N subsets according to their destinations. The decision of accepting a request or not in one subset does not affect the decisions in other subsets, as no connection request belongs to two subsets. This provides a basis for a fast distributed scheduling algorithm, as the scheduling for each output fiber can be done independently of other output fibers. As will be seen later, the time complexity of the distributed scheduling algorithms we propose is only in terms of the number of wavelengths per fiber and the conversion degree, and is independent of the interconnect size. A global scheduling algorithm, on the other hand, will have a time complexity at least linear to the size of the interconnect. Thus, in the following we will focus on the scheduling algorithms that have a distributed nature and can be run independently for each output fiber. The input to the scheduling algorithm is the connection requests destined to this fiber. The output of the algorithm is the decision whether a request is granted or not, and if granted, which wavelength channel it is assigned to. Note that if the wavelength conversion is full range the scheduling is trivial: if no more than k connection requests arrived at this output fiber, grant all; if more than k arrived, arbitrarily pick k out of them. This is because full range wavelength converters are capable of converting any incoming wavelength to any outgoing wavelength, which makes the connection requests indistinguishable in the wavelength domain.

However, the scheduling becomes more complex when we use limited range wavelength converters, as the wavelengths of connection requests can no longer be considered indistinguishable. For example, in the interconnect in Figure 1, if $k = 6$, and $d = 3$, consider the case where two connections on λ_1, three connections on λ_2 and one connection on λ_4 have arrived and all want to destine to the same output fiber with six wavelengths $\lambda_0, \lambda_1, \ldots, \lambda_5$. If the wavelength conversion is full range, all of them can be satisfied because there are totally six requests which does not exceed the total number of wavelengths on a fiber. However, since the conversion degree is 3 in this interconnect, not all can be satisfied. This is because that there are five connection requests arrived on λ_1 and λ_2, but there are only four output wavelengths accessible to λ_1 and λ_2: $\{\lambda_0, \lambda_1, \lambda_2, \lambda_3\}$ with conversion degree 3. If not all connection requests can be satisfied, we say there is *output contention* among the connection requests. When a contention occurs, the scheduling algorithm needs to select some of the connection requests to transmit while reject others. To maximize network throughput, the algorithm should find the largest group of requests that are contention-free. In this example, we can drop one request coming on λ_1 or λ_2 and realize the rest of connection requests.

Extensive research has been conducted on scheduling algorithms for various electronic switches (which can be considered as a single wavelength switch). For example, [7] and [8] considered scheduling algorithms in input-buffered electronic switches under unicast traffic. Scheduling algorithms for WDM broadcast and select networks were also well studied in recent years, see, for example, [17], [18]. In this type of network, the source node broadcasts its information to all other nodes via a selected wavelength, and only the destination node tunes into this wavelength to get the message, so that only one wavelength on the fiber is used at a time, both for the source and the destination node. Note that this is a quite different type of network from the WDM interconnect considered in this paper. We consider a space-division switch where all wavelengths on a fiber can be utilized simultaneously. There has also been some work in the literature on the performance analysis of WDM optical interconnects with limited range wavelength conversion in WDM wavelength routing networks, e.g., [11], [13], [14]. In these studies, the packet arrivals at the optical interconnect were assumed to be asynchronous, thus eliminates the need for a scheduling algorithm since the requests have a natural order and are assumed to be served according to the "first come first served" rule. However, future WDM optical interconnects are more likely to be operated synchronously [9] [10], for example, optical packet switching networks [12], where connection requests arrive in time slots. This type of interconnect is also more suitable for interconnecting processors in a parallel or distributed computing system as synchronization is required among collaborating processors. In such an interconnect, a general method for resolving output contention has to be studied. In this paper, we will focus on contention-free scheduling algorithms for such synchronized WDM optical interconnects.

II. Preliminaries

A. Wavelength Conversion

As mentioned earlier, in a WDM optical interconnect with limited range wavelength conversion, an incoming wavelength may be converted to a set of adjacent outgoing wavelengths, and the number of wavelengths convertible is equal to the conversion degree. We can use a *conversion graph* to visualize the wavelength conversion. A conversion graph is a bipartite graph, in which each of the left side vertex represents an input wavelength and each of the right side vertex represents an output wavelength. Thus, if there are k wavelengths on each fiber, there are k vertices on each side of the conversion graph. If input wavelength λ_i can be converted to output wavelength λ_j, we draw an edge between them. Figure 2(a) is the conversion graph for $k = 6$ and conversion degree $d = 3$, in which, wavelength λ_i may be converted to $\{\lambda_{(i-1) \mod 6}, \lambda_i, \lambda_{(i+1) \mod 6}\}$, for $i \in \{0, 1, \ldots, 5\}$.

In general, we assume λ_i may be converted to $\{\lambda_{(i-e) \mod k}, \lambda_{(i-e+1) \mod k}, \ldots, \lambda_i, \ldots, \lambda_{(i+f) \mod k}\}$, where e and f are the numbers of wavelengths that λ_i may be converted to on its "minus" and "plus" side, respectively. Clearly, $e+f+1 = d$. In the example above $e = f = 1$. We call the wavelengths that λ_i may be converted to as the *adjacency set* of λ_i in the conversion graph, and represent the adjacency set as an interval of integers: $[i-e, i+f]$. Note that all the numbers inside this interval are in "mod k". In other words, interval $[x, y]$ represents numbers $\{x \mod k, (x+1) \mod k, \ldots, y \mod k\}$. This notation is purely for presentational convenience, since the adjacency set of some wavelength may not be an interval. For example, the adjacency set of λ_0 is $\{\lambda_5, \lambda_0, \lambda_1\}$ and apparently $5, 0$, and 1 are not an interval, but since $(5 \mod 6) = (-1 \mod 6)$, we can represent it as $[-1, 1]$.

The conversion scheme discussed above is the common assumption about limited range wavelength conversion in the literature. The resulting conversion graph is "circular symmetrical". There are also other types of assumptions about wavelength conversion which is not circular symmetrical. For example, some authors considered the adjacency set of any λ_i is interval $[u, v]$, where $u = \max\{0, i-e\}$ and $v = \min\{k-1, i+f\}$, which means that the wavelengths near one end cannot be converted to the wavelengths on the other end. For example, if $k = 6$ and $e = f = 1$, λ_0 can only be converted to λ_0 and λ_1, and it cannot be converted to λ_5. Figure 2(b) shows the conversion graph of this type of conversion when $k = 6$ and $e = f = 1$. Note that the adjacency set of this type of conversion is indeed an interval. We will consider both types of wavelength conversions in this paper.

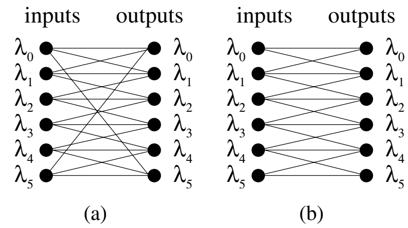

Fig. 2. Conversion graphs of two types of wavelength conversion six wavelengths and conversion degree three. (a) Circular symmetrical. (b) Non-circular symmetrical.

B. Problem Formalization

As described in Section I, we consider an $N \times N$ WDM optical interconnect with k wavelength on each fiber. Any input wavelength channel can be connected to its adjacency set on any output fiber according to its conversion degree. We depict the relationship between the connection requests destined for an output fiber and the available wavelength channels on that output fiber by a bipartite graph, called *request graph*. On the left side of the request graph, each node represents a connection request and on the right side of the graph, each node represents an output wavelength. We use A to represent the set of the left side vertices and B for the right side vertices. Let the vertices on the right side be in the same order as their corresponding wavelength indexes. For example, λ_0 is above λ_1, λ_1 is above λ_2 and so on. The vertices on the left side are also ordered according to their wavelength indexes (requests with the same wavelength are in an arbitrary order). There is an edge between a left side node a and a right side node b if the wavelength of connection request a can be converted to output wavelength b. Thus the conversion graph discussed earlier can be simply considered as a special case of the request graph when there is exactly one connection request coming on each wavelength. For convenience we also define the *request vector*. A request vector is a $1 \times k$ row vector, with the i_{th} element representing the number of connection requests arrived on wavelength λ_i. Figure 3 shows the request graphs of the two types of conversions when the request vector is $[2, 1, 0, 1, 1, 2]$. In addition, for any $a_i \in A$, let $W(i)$ denote the wavelength index of connection request a, $W(i) \in [0, k-1]$. For example, in Figure 3(a), $W(0) = W(1) = 0$, and $W(2) = 1$. When no confusions arise, we will use it to represent the wavelength of connection request a_i as well.

A request graph of an output fiber can be easily formed and stored in hardware. For example, the left side vertices of the request graph can be implemented by an $Nk \times 1$ binary vector

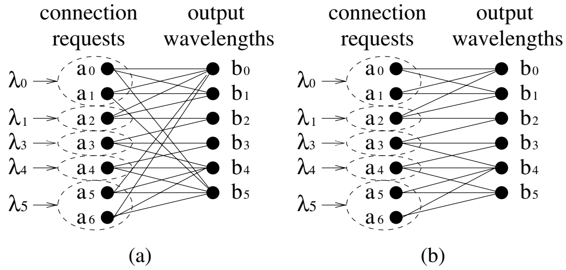

Fig. 3. Request graphs of two types of conversions when the request vector is $[2, 1, 0, 1, 1, 2]$ in a 6-wavelength interconnect with conversion degree 3. (a) Circular symmetrical. (b) Non-circular symmetrical.

(an Nk bit register), with element $(i-1)*k+j$ being 1 means λ_j on the i_{th} input fiber is destined for this output fiber and 0 otherwise. The $Nk \times 1$ binary vector is set at the beginning of each time slot. The right side vertices of the request graph can be implemented by a $k \times 1$ vector with each element storing the decision of which input wavelength channel it is assigned to. The edges that represent the relationship between the inputs and outputs do not need to be implemented explicitly and can be considered embedded in the circuit.

In a request graph G, let E denote the set of edges. Any wavelength assignment can be represented by a subset of E, E_1, where edge $ab \in E_1$ if wavelength channel b is assigned to connection request a. Under unicast traffic, any connection request needs only one output channel and an output channel can be assigned to only one connection request. It follows that the edges in E_1 are vertex disjoint, because that if two edges share a vertex, either one connection request is assigned two wavelength channels or one wavelength channel is assigned to two connection requests. Thus, E_1 is a *matching* in G. For a given set of connection requests, to maximize network throughput, we should find a maximum matching in the request graph. In the example above, not all left side vertices can be matched because there are seven vertices on the left (connection requests) while there are only six vertices on the right (available wavelength channels). The maximum matchings for both types of conversions are shown in Figure 4, and in this case they are identical.

The best known algorithm for finding maximum matching in an arbitrary bipartite graph was given in [1], and has time complexity $O(n^{\frac{1}{2}}(m+n))$, where n and m are the number of vertices and edges in the bipartite graph, respectively. If we directly adopt this algorithm in our scheduling algorithm, the time complexity would be as high as $O(N^{\frac{3}{2}}k^{\frac{3}{2}}d)$, since the left side vertices in a request graph alone could be as large as Nk and each left side vertex is adjacent to d right side vertices. However, faster algorithms are required for scheduling in WDM optical interconnects as the decision has to be made in real-time within a time slot, which is in the order of μs [12]. In the rest of the paper, we will show that the request graph for limited range wavelength conversion exhibits some nice properties so that faster algorithms are possible. We will present two fast scheduling algorithms with time complexity $O(k)$ and $O(dk)$, respectively, for two types of wavelength conversions, where k is the number of wavelengths and d $(\leq k)$ is the conversion degree. Note that the time complexity of our algorithms is independent of the interconnect size N, and can be easily implemented in hardware also. Because finding the largest group of contention-free connection requests is equivalent to finding a maximum matching in the request graph, in the following we will describe these algorithms in the form of finding a maximum matching in bipartite graphs. We first consider the case that all output wavelength channels are available at the beginning of time slots in Section III and Section IV. Then in Section V, we show that our algorithms can be easily extended to the case that some of the output wavelength channels are occupied.

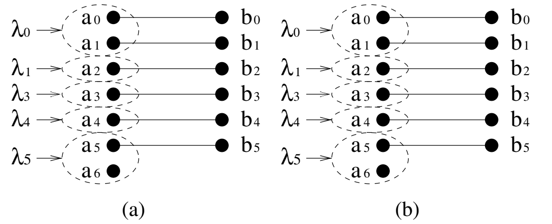

Fig. 4. Maximum matchings in two request graphs in Figure 3. (a) Circular-symmetrical. (b) Non-circular-symmetrical.

III. SCHEDULING ALGORITHM FOR NON-CIRCULAR SYMMETRICAL WAVELENGTH CONVERSION

First we consider non-circular symmetrical wavelength conversion. In this case, the request graph is a *convex bipartite graph* as defined in [2]. A bipartite graph G is convex if there exists an ordering "$\leq$" of the right side vertices B such that for any $a \in A$ and distinct $b_1, b_2 \in B$ (with $b_1 \leq b_2$), edge $ab_1 \in E$ and $ab_2 \in E \Rightarrow ab \in E$ for any $b \in B$ and $b_1 \leq b \leq b_2$, where E is the set of edges in G. In other words, if we use $B(a)$ to denote the set of vertices in B adjacent to a, $B(a)$ is an interval for any $a \in A$ in this ordering. With non-circular symmetrical wavelength conversion, a request graph is convex because for every left side node a, $B(a)$ is the set of wavelengths that wavelength $W(a)$ may be converted to, and they form an interval if the right side vertices are ordered according to the wavelength indexes. For example, in Figure 3(b), if we check left side vertex a_2, the set of vertices adjacent to it is $B(a_2) = \{b_0, b_1, b_2\}$, which can be represented by interval $[0, 2]$. And so do other left side vertices.

Simpler algorithms exist for finding a maximum matching in a convex bipartite graph. The algorithm shown in Table 1 was described in [3]. The input to this algorithm is: (1) The left side vertex set A and the right side vertex set B; (2) For each left side vertex a, the set of vertices adjacent to it denoted by interval $[BEGIN(a), END(a)]$. We call $BEGIN(a)$ and $END(a)$ the BEGIN value and END value of a, respectively. The output of the algorithm is array $MATCH[]$. $MATCH[i] = j$ means that the i_{th} right side vertex is matched to the j_{th} left side vertex. $MATCH[i] = \Lambda$ if the i_{th} right side vertex is not matched to any left side vertex. We can see that in this algorithm, the i_{th} vertex in B is

TABLE 1

GLOVER'S ALGORITHM FOR FINDING A MAXIMUM MATCHING IN A CONVEX BIPARTITE GRAPH

```
Glover's Algorithm
for i := 1 to | B | do
    U:={k : k ∈ A, (i,k) ∈ E}
    if U = ∅
        MATCH[i] := Λ
    else
        j := element in U with minimum END value
        MATCH[i] := j
        delete j from A
    end if
end for
```

TABLE 2

FIRST AVAILABLE ALGORITHM FOR FINDING A MAXIMUM MATCHING IN A REQUEST GRAPH OF NON-CIRCULAR SYMMETRICAL WAVELENGTH CONVERSION

```
First Available Algorithm
for i := 1 to k do
    let a_j be the first vertex in A adjacent to b_i
    if no such a_j exists
        MATCH[i] := Λ
    else
        MATCH[i] := j
        delete a_j from A
    end if
end for
```

matched to an adjacent vertex in A whose interval ends closest to it. The algorithm has time complexity $O(|E|) = O(Nkd)$, and the time consuming part is the formation of set U and finding the smallest-ending vertex j.

Glover's algorithm has a lower time complexity than the algorithm for general bipartite graphs in [1] since the bipartite graph is convex. Our request graph is a special type of convex bipartite graphs. Hence Glover's algorithm can be further simplified into First Available Algorithm as described in Table 2. In this algorithm, we match right side vertex b_i with the first left side vertex that is adjacent to it. In other words, we choose the "top" edge in the request graph and add it to the matching in each iteration. Notice that here, for a right side vertex, instead of finding all the adjacent left side vertices and then selecting among them the one with the smallest END value, all we need to do is to find the first adjacent left side vertex. We have the following result concerning this algorithm.

Theorem 1: First Available Algorithm finds a maximum matching in a request graph of non-circular symmetrical wavelength conversion.

Proof. In a request graph of non-circular symmetrical wavelength conversion, we notice that for left side vertices a_j and a_l, if $j \leq l$, then $BEGIN(j) \leq BEGIN(l)$ and $END(j) \leq END(l)$. If we use Glover's Algorithm to find the maximum matching for the request graph, in each iteration, when we are searching a matching for the i_{th} right side vertex, we can simply match it to the vertex with smallest index on the left side that is adjacent to it. This is because that the END value of any other vertex with larger index cannot be smaller than it. Thus, Glover's Algorithm can be simplified to First Available Algorithm when applied to a request graph of non-circular symmetrical wavelength conversion.

∎

Now we discuss the implementation of this algorithm. From Table 2, we should find the first adjacent vertex to a right side vertex in each step. Since the packets on the same wavelength are identical for the purpose of maximizing the matching size, we need only to find the first input wavelength that has at least one packet and can be converted to current output wavelength. After that if there are more than one packets on this input wavelength, to ensure fairness, a random selecting or a round-robin scheduling procedure should be adopted as suggested in [7] [8]. All this can be implemented in hardware and the execution time of each step would be a constant. Thus, the time complexity of this algorithm is $O(k)$. Note that the time complexity is independent of the interconnect size and conversion degree. Of course, if less complex hardware is used it may not be the case. However, since timing is critical here, the method described above is preferred.

IV. SCHEDULING ALGORITHM FOR CIRCULAR SYMMETRICAL WAVELENGTH CONVERSION

When wavelength conversion is circular symmetrical, the request graph is no longer a convex bipartite graph, because that the left side vertices near one end will have links to the right side vertices on the other end (e.g., edge a_0b_5 and a_6b_0 in Figure 3), and the adjacency set is not an interval. This makes the problem more complicated. However, in the following discussion we show that we can still apply the First Available Algorithm by "breaking" the request graph.

A. Breaking the request graph

First, we introduce some useful definitions in our discussion.

Definition 1: Edge a_jb_v *crosses* a_ib_u when

Case 1. $W(j) \neq W(i)$

1.1. $W(j) \in [u-f+1, W(i)-1]$ and $v \in [u+1, W(j)+f]$

1.2. $W(j) \in [W(i)+1, u-1+e]$ and $v \in [W(j)-e, u-1]$

Case 2. $W(j) = W(i)$

2.1. $j < i$ and $v \in [u+1, W(j)+f]$

2.2. $j > i$ and $v \in [W(j)-e, u-1]$

For example, in Figure 3(b) edges a_0b_1 and a_1b_0 cross each other, edge a_3b_4 crosses a_4b_3, but edge a_0b_5 and a_4b_4, though intersecting with each other in the figure, are not a pair of crossing edges.

Definition 2: For request graph G, let a_i and b_u be two vertices such that $a_ib_u \in E$. Let G' be the subgraph of G obtained by removing vertices a_i and b_u, all the edges incident to them, and all edges that cross edge a_ib_u in G. We call G' the *reduced graph* of G. By doing this, we say we *break* G at a_ib_u. Edge a_ib_u is called the *breaking edge*.

We have the following result.

Lemma 1: There exists a maximum matching in the request graph where there are no crossing edges.

Proof. We prove this lemma by showing that every pair of crossing edges in the maximum matching can be replaced by two non-crossing edges. Let G be the request graph and E be the edge set of G. Suppose edge a_ib_u and a_jb_v are in a maximum matching M and cross each other. If $W(j) = W(i)$, when edge $a_ib_u \in E$ and $a_jb_v \in E$, we have $a_ib_v \in E$ and $a_jb_u \in E$. Edge a_ib_v and a_jb_u are not in M, because if one of them is then there would be two edges sharing one vertex in M. They also do not cross each other. In Case 1.1 when $j < i$, from the definition we know that $v \in [u+1, W(j)+f]$, thus $[v+1, W(j)+f]$ is a subset of $[u+1, W(j)+f]$. But $u \notin [u+1, W(j)+f]$, then $u \notin [v+1, W(j)+f]$. Thus, by definition a_ib_v and a_jb_u do not cross each other. A similar proof can be given for Case 1.2. Therefore, we can remove a_ib_u and a_jb_v from M and add a_ib_v and a_jb_u to M. The new matching is still a maximum matching.

For $W(j) \neq W(i)$, we prove Case 2.1 when $W(j) \in [u-f+1, W(i)-1]$ and $v \in [u+1, W(j)+f]$. The proof for Case 2.2 is similar. Since $a_ib_u \in E$, the set of left side vertices with wavelengths $[u-f, W(i)]$ are adjacent to b_u. Also, since $W(j) \in [u-f+1, W(i)-1]$, $a_jb_u \in E$. We now show that $a_ib_v \in E$. The adjacency set of wavelength $W(i)$ is $I = [W(i)-e, W(i)+f]$. If $(W(j)+\sigma) \mod k = W(i)$, the adjacency set of wavelength $W(j)$ is $J = [W(i)-e-\sigma, W(i)+f-\sigma]$. By definition, $v \in [u+1, W(j)+f] = [u+1, W(i)+f-\sigma] \subset [u, W(i)+f]$. Since $u \in I$ and $[u, W(i)+f] \subseteq I$, it follows that $v \in I$, or, $a_ib_v \in E$. Edge a_ib_v does not cross a_jb_u because $W(j) \in [v-f+1, W(i)-1]$ but $u \notin [v+1, W(j)+f]$. Hence, for $W(j) \neq W(i)$ the pair of crossing edges can also be replaced by two non-crossing edges not in M. We can apply the same procedure to every pair of crossing edges in M and obtain a new maximum matching without crossing edges. ∎

As an example, in Figure 3(b), edge a_0b_1 and a_1b_0 are a pair of crossing edges. If they are in a matching, we can replace them with edge a_0b_0 and a_1b_1. Similarly, edge a_3b_4 and a_4b_3 are a pair of crossing edges. If they are in a matching, we can replace them with edge a_3b_3 and a_4b_4.

For left side vertex a_j, the adjacency set in G can be represented as $[W(j)-e, W(j)+f]$. When G is broken at edge a_ib_u, a_j's adjacency set in G' will be in one of the following three forms: $[W(j)-e, u-1]$, $[u+1, W(j)+f]$, or $[W(j)-e, W(j)+f]$. This is because when we break G at edge a_ib_u, if $W(j) \in [u-f+1, W(i)-1]$, the links to wavelengths $[u+1, W(j)+f]$ would have been deleted and the adjacency set in G' is $[W(j)-e, u-1]$. If $W(j) \in [W(i)+1, u-1+e]$, the links to wavelengths $[W(j)-e, u-1]$ would have been deleted and the adjacency set in G' is $[u+1, W(j)+f]$. If $W(j) = W(i)$, when $j > i$, the adjacency set in G' is $[u+1, W(i)+f]$; when $j < i$, the adjacency set in G' is $[W(i)-e, u-1]$. Otherwise $W(j) \notin [u-f, u+e]$, i.e., a_j is not adjacent to b_u in G, and the adjacency set in G' is still $[W(j)-e, W(j)+f]$.

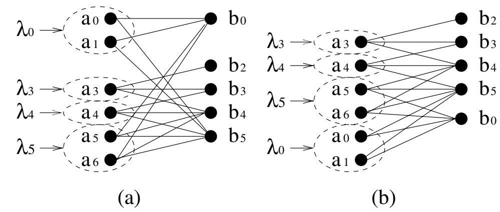

Fig. 5. Breaking at edge a_2b_1 in Figure 3(a) (a) Deleting vertex a_2 and b_1 and all the incident edges and crossing edges. (b) Reordering a_3 and b_2 to the top.

Now when G is broken, the order of vertices in G' can be represented as:

$$a_0, a_1, \dots, a_{i-1}, a_{i+1} \dots, a_{|A|-1}, \text{ and}$$
$$b_0, b_1, \dots, b_{u-1}, b_{u+1}, \dots, b_{|B|-1}$$

We now left shift these vertices until a_{i+1} and b_{u+1} are at the left end:

$$a_{i+1}, a_{i+2}, \dots, a_{|A|-1}, a_0, a_1, \dots, a_{i-1}, \text{ and}$$
$$b_{u+1}, b_{u+2}, \dots, b_{|B|-1}, b_0, b_1, \dots, b_{u-1}.$$

In the above shifting, only the ordering of vertices is changed, but no edges are deleted or added. It is not difficult to verify that all three types of adjacency sets are intervals in the new ordering: $[u+1, W(j)+f]$ represents the left most $W(j)+f-u$ vertices, $[W(j)-e, u-1]$ represents the right most $u-W(j)+e$ vertices, and $[W(j)-e, W(j)+f]$ represents d consecutive vertices in the middle. Hence G' in is a convex bipartite graph. Moreover, for vertices a_j and a_l, if vertex a_j appears on the left of a_l in the new ordering, the first and the last vertex adjacent to a_j will also be on the left of the first and the last vertex adjacent to a_l, respectively. This means that in the new vertex ordering of G', for left side vertices a_j and a_l, if $j \leq l$, then $BEGIN(j) \leq BEGIN(l)$, $END(j) \leq END(l)$. Thus we have:

Lemma 2: The vertices in the reduced graph G' can be ordered in a way such that the First Available Algorithm can be used to find the maximum matching for G'.

Figure 5 shows an example of breaking the request graph in Figure 3(a) at edge a_2b_1.

Now the question is whether a maximum matching in G' along with the breaking edge is a maximum matching of G. The following lemma shows this is true if the breaking edge is in a no-crossing-edge maximum matching of G.

Lemma 3: If edge a_ib_u is in a no-crossing-edge maximum matching M of request graph G, then edge a_ib_u along with a maximum matching of graph G' which is obtained by breaking G at edge a_ib_u is a maximum matching of G.

Proof. If edge a_ib_u is in a no-crossing-edge maximum matching M of G, then edges in $L = M \setminus \{a_ib_u\}$ are all in G'. It follows that L is a matching of G'. It is also true that any matching of G' along with edge a_ib_u is a matching of G. Now L is a maximum matching of G', since if it is not, the maximum

TABLE 3

BREAK AND FIRST AVAILABLE ALGORITHM FOR FINDING A MAXIMUM MATCHING IN A REQUEST GRAPH OF CIRCULAR SYMMETRICAL WAVELENGTH CONVERSION

Break and First Available Algorithm
arbitrary choose one left side vertex a_i
do for all right side vertex b_u adjacent to a_i
begin
break G at edge $a_i b_u$, let the reduced graph be G_u.
apply First Available Algorithm to G_u
to find a maximum matching M_u.
end
let M_w be the matching with max cardinality among all M_u
return $M_w \cup a_i b_u$

matching of G' written as L' contains more edges than L. Then $L' \cup \{a_i b_u\}$ which is also a matching of G contains more edges than M. This contradicts with the fact that M is a maximum matching. Therefore, any maximum matching of G' contains the same number of edges as L. Hence, any maximum matching of G' together with edge $a_i b_u$ contains the same number of edges as M. The lemma follows. ∎

Finding an edge in G that is in a no-crossing-edge maximum matching is not trivial. Examples can be found that none of the d edges incident to a left side vertex can be assumed to be in a no-crossing-edge maximum matching. However, we can prove that there must be at least one such an edge among the d edges.

Lemma 4: For any left side vertex a_i in the request graph G, at least one of the d edges incident to it is in some no-crossing-edge maximum matching.

Proof. We say a_i is saturated in matching M if there is an edge in M incident to a_i. It is obvious that there must be a maximum matching in which a_i is saturated. Since given any maximum matching M', if a_i is saturated, the claim is true. If a_i is not saturated in M', all the right side vertices adjacent to a_i must be saturated. Now we can arbitrarily match a_i to one of them, say, b_u, and remove the edge in M' incident to b_u, By doing this we obtain a new maximum matching in which a_i is saturated. Now suppose M'' is a maximum matching in which a_i is saturated. If M'' is already a no-crossing-edge matching, then M'' is the maximum matching we need. If there are crossing edges in M'', we can apply the procedure described in Lemma 1 to obtain a no-crossing-edge maximum matching M. Notice that any vertex saturated in M'' is still saturated in M. The lemma follows. ∎

From this lemma we know that if we try all d edges, we can obtain a maximum matching.

B. The Scheduling Algorithm

Now we are in the position to present the Break and First Available Algorithm as described in Table 3.

By combining Lemmas 2, 3 and 4, we can obtain the following result concerning the Break and First Available Algorithm.

Theorem 2: The Break and First Available Algorithm gives the maximum matching of a request graph of circular symmetrical wavelength conversion.

The time complexity of the Break and First Available Algorithm is $O(dk)$, because we need to try all d reduced graphs, and each reduced graph has $k-1$ right side vertices. Again note that the time complexity is independent of network size N. We can also implement this algorithm in parallel and time complexity could be reduced to $O(k)$, but we then need d units of hardware. If the conversion degree d is large, the time complexity or the hardware cost will be high. However, as we mentioned in Section I, in WDM networks, it has been shown that with very small conversion degree, the network performance can be close to the case of full conversion ($d = k$), [11]. Most of the proposed WDM optical interconnects with limited range wavelength conversion have a conversion degree of $d = 2$ or $d = 3$. Thus, Break and First Available Algorithm is suitable for hardware implementation in this case.

C. Discussions on the Approximation Algorithm

In the Break and First Available Algorithm we try all d reduced graphs because we do not know before hand which edge belongs to a no-crossing-edge maximum matching. In some applications, if the speed of the scheduling algorithm is of more concern than achieving the maximum network throughput, we can trade-off the time complexity of the algorithm with the network throughput (the size of the matching). Now we discuss such an approximation algorithm. The question is: if we want to try only one reduced graph to save time or hardware cost, which reduced graph we should choose such that the matching we obtain will still be close to a maximum matching.

Lemma 5: If edges $a_j b_v$ and $a_l b_w$ crosses edge $a_i b_u$ and $W(j) \in [W(i)+1, u-1+e]$, $W(l) \in [u-f+1, W(i)-1]$, then edge $a_j b_v$ and $a_l b_w$ cross each other.

Proof. We have $v \in [W(j)-e, u-1]$ and $w \in [u+1, W(l)+f]$. Hence $v \in [W(j)-e, w-1]$. If $W(j) \in [W(l)+1, w-1+e]$, edge $a_j b_v$ and $a_l b_w$ cross each other. $W(j) \in [W(i)+1, u-1+e]$ and $W(l) \in [u-f+1, W(i)-1]$ imply that $W(j) \in [W(l)+1, u-1+e]$. Similarly, since $w \in [u+1, W(l)+f]$, we have $W(j) \in [W(l)+1, w-1+e]$. ∎

Now the d right side vertices adjacent to a_i are $\{W(i)-e, W(i)-e+1, \ldots, W(i), \ldots, W(i)+f\}$. If edge $a_i b_u \in E$, then u is in this set. Suppose u is the $\delta(u)_{th}$ element in the set when counted from the left to the right. For example, if $u = (W(i)-e) \mod k$, then $\delta(u) = 1$; if $u = (W(i)-e+1) \mod k$, then $\delta(u) = 2$. We have the following lemma.

Lemma 6: Edge $a_i b_u$ may cross up to $\max\{\delta(u)-1, d-\delta(u)\}$ edges in a no-crossing-edge maximum matching.

Proof. Given a no-crossing-edge maximum matching M, if $a_i b_u \in M$, it does no cross any edge in M. If $a_i b_u \notin M$, we first consider the requests coming in the wavelength range $[u-f+1, W(i)]$. The set of right side vertices adjacent to them and that may constitute a crossing edge of $a_i b_u$ can be represented as $[u+1, W(i)+f]$. There are totally $d-\delta(u)$ elements in this set. It follows that in any matching there are at most this number of edges crossing $a_i b_u$ for the left side vertices in the wavelength range $[u-f+1, W(i)]$. By a similar argument, there are at most $\delta(u)-1$ edges crossing a_ib_u with left side vertices in the wavelength range $[W(i), u-1+e]$. From Lemma 5, any edge in one group crosses all the edges in the other group. Thus, if the matching has no crossing edges, only one group may be in that matching. Hence, the maximum number of edges crossing a_ib_u is the larger of the two. ∎

Theorem 3: By breaking the request graph at edge a_ib_u, the difference between the matching we find and a maximum matching cannot be greater than $\max\{\delta(u)-1, d-\delta(u)\}$, where b_u is the $\delta(u)_{th}$ vertex adjacent to a_i when counting from the end of the "minus" side.

Proof. Given any no-crossing-edge maximum matching M, by breaking the request graph at edge edge a_ib_u, G' contains at least $\mid M \mid - \max\{\delta(u)-1, d-\delta(u)\} - 1$ edges in M. These edges are also a matching in G'. Hence the cardinality of the maximum matching of G' cannot be less that that. Thus by breaking the request graph at edge a_ib_u we find a matching with at least $\mid M \mid - \max\{\delta(u)-1, d-\delta(u)\}$ edges. ∎

Corollary 1: The upper bound on the difference between a maximum matching of G and the matching given by breaking G at edge a_ib_u achieves the smallest value $(d-1)/2$ when $\delta(u) = (d+1)/2$.

From this Corollary, if $e = f$, $\delta(u) = (d+1)/2$ means we should choose the "shortest" edge. For the case of $d = 3$ ($e = f = 1$), by breaking the request graph at the "shortest" edge, the difference between our matching and a maximum matching is at most 1. For the case of $d = 5$ ($e = f = 2$), the difference is at most 2.

V. Extensions

In the previous discussions we consider the case that all k output channels $\lambda_0, \lambda_1, \dots, \lambda_{k-1}$ are available at the time of scheduling. In this case, in the request graph there are always k vertices on the right side of a request graph. However, if the duration of connections is more than one time slot it may occur that not all of the output channels are available at the time of scheduling, since some output channels may still be occupied by previously arrived connection requests. If the existing connections can be disturbed, i.e., be reassigned to a different output channel if needed, the algorithms discussed in the previous section can still be used. If it is not the case, which is true in optical burst switching, we can redraw the request graph by removing the right side vertices representing those occupied wavelength channels and all the edges incident to them. It is not difficult to see that we can still apply the algorithms introduced previously to these modified request graphs to obtain the maximum matching.

VI. Conclusions

In this paper we have studied scheduling algorithms to resolve output contentions in WDM optical interconnects with wavelength conversion ability. We presented the solutions for the general case of limited range wavelength conversion, including the full range wavelength conversion. We have introduced the request graph and showed that the largest group of contention-free connection requests can be found by finding a maximum matching in the request graph. We studied two types of limited range wavelength conversions, circular symmetrical and non circular symmetrical. We showed that the request graphs under these types of wavelength conversions have some nice properties and presented fast, simple optimal algorithms which always find a maximum matching to achieve maximum network throughput. Compared with the general algorithm for finding a maximum matching in bipartite graphs with time complexity $O(N^{\frac{3}{2}}k^{\frac{3}{2}}d)$ [1], the distributed algorithms we gave have time complexity of $O(k)$ and $O(dk)$ (independent of interconnect size N) for non-circular symmetrical and circular symmetrical wavelength conversion, respectively, where k is the number of wavelengths and d is the conversion degree. In addition, our algorithms can be easily implemented in hardware, and used for time slotted WDM optical interconnects where connections hold for different number of time slots. Interesting future work may include incorporating different QoS requirements, such as different priorities among connection requests, in the scheduling algorithm.

References

[1] J. Hopcroft and R. Karp "An $n^{\frac{5}{2}}$ algorithm for maximum matchings in bipartite graph," *SIAM J. Comput.*, 2(4):225-231, Dec. 1973.

[2] F. Glover "Maximum matching in convex bipartite graph," *Naval Res. Logist. Quart.*,14, pp. 313-316, 1967.

[3] W. Lipski Jr and F.P. Preparata "Algorithms for maximum matchings in bipartite graphs," *Naval Res. Logist. Quart.*,14, pp. 313-316, 1981.

[4] B. Mukherjee, "WDM optical communication networks:progress and challenges," *JSAC*, vol. 18, no. 10, pp. 1810-1824, 2000.

[5] D. K. Hunter, M. C. Chia and I. Andonovic "Buffering in optical packet switches," *J. Lightwave Tech.*, vol. 16 no. 12, pp. 2081-2094, 1998.

[6] M. Kovacevic and A. Acampora, "Benefits of wavelength translation in all-optical clear-channel networks," *JSAC*, vol. 14, no. 5, pp. 868-880, 1996.

[7] N. McKeown, P. Varaiya, and J. Warland, "Scheduling cells in an input-queued switch," *IEEE Eletron. Lett.*, pp. 2174-2175, 1993.

[8] N. McKeown, "The iSLIP scheduling algorithm input-queued switch," *IEEE/ACM Trans. Networking*, vol. 7, pp. 188-201, 1999.

[9] H. J. Chao, C. H. Lam and E. Oki, *Broadband Packet Switching Technologies*, Wiley-Interscience, 2001.

[10] Walter J. Goralski, *Optical Networking and WDM*, McGraw-Hill, 2001.

[11] T. Tripathi and K. N. Sivarajan, "Computing approximate blocking probabilities in wavelength routed all-optical networks with limited-range wavelength conversion," *JSAC*, vol. 18, pp. 2123–2129, 2000.

[12] L.Xu, H. G. Perros and G. Rouskas, "Techniques for optical packet switching and optical burst switching," *IEEE communications Magazine*, pp. 136-142, Jan. 2001.

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

[14] X. Qin and Y. Yang, "Nonblocking WDM switching networks with full and limited wavelength conversion," *IEEE Trans. Comm.*, vol 50, no. 12, 2002.

[15] Y. Yang and J. Wang, "WDM optical interconnect architecture under two connection models," *Proc. of IEEE Hot Interconnects 10, Symposium on High Performance Interconnects*, pp. 146-151, Palo Alto, CA, 2002.

[16] Y. Yang, J. Wang and C. Qiao "Nonblocking WDM multicast switching networks," *IEEE Trans. Parallel and Distributed Systems*, vol. 11, no. 12, pp. 1274-1287, 2000.

[17] M.S. Borella and B. Mukherjee "Efficient scheduling of nonuniform packet traffic in a WDM/TDM local lightwave network with arbitrary transceiver tuning latencies," *JSAC*, vol. 14, no. 5, pp. 923-934, 1996.

[18] G.N. Rouskas and V. Sivaraman "Packet scheduling in broadcast WDM networks with arbitrary transceiver tuning latencies," *IEEE/ACM Trans. Networking*, vol. 14, no. 3, pp. 359-370, 1997.