A Simple Routing and Wavelength Assignment Algorithm using the Blocking Island Technique for All-Optical Networks [1]

Ding Zhemin and Mounir Hamdi

Department of Computer Science, Hong Kong University of Science and Technology
Clear Water Bay, Kowloon, Hong Kong

Abstract-- **In this paper we consider the routing and wavelength assignment problem as well as the placement of wavelength converters in a wavelength routed all-optical network. Using a clustering technique called BI (*Blocking Island*), we propose a simple and intelligent RWA (*Routing and Wavelength Assignment*) algorithm: BI_RWA and a converter placement algorithm. These algorithms can be used in arbitrarily connected networks and with some simple modifications, they can also be applied on various networking scenarios. We have evaluated our algorithms through extensive simulations. The simulations are carried out in two parts: static traffic and dynamic traffic. The results will demonstrate that our RWA algorithm performs better than other previously proposed algorithms (in the cases we studied).**

Index Terms-- **Routing and Wavelength Assignment, Blocking Island, All-Optical networks**

I. INTRODUCTION

In an optical network with wavelength-division multiplexing (WDM) transmission, each data channel is carried on a unique wavelength (or optical frequency) and a single optical fiber has many different wavelengths. With the development of optical cross-connects and WDM technology, high-speed, end to end connections called lightpaths can be routed from sources to destinations, simplifying network management and processing. Networks which use optical cross-connects to route lightpaths through the network are referred to as wavelength routed networks. In a wavelength routed WDM network, a lightpath (e.g., wavelength continuous path without processing in the intermediate nodes) is established between two communication nodes. A lightpath may span multiple fiber links and must occupy the same wavelength on all the fiber links it traverses if there are no wavelength converters. This property is known as the wavelength continuity constraint. In order to satisfy a lightpath request in wavelength routed WDM networks, we not only need to consider the routing but the wavelength selection as well. Given a set of connection requests, the problem of setting up a lightpath by routing and assigning a wavelength to each connection is called the RWA problem. It can be formulated as a combinatorial problem which is known to be NP-complete [2]. As a result, various heuristic algorithms have been proposed and evaluated under different networking assumptions. The traffic assumptions generally fall into one of the two categories: 1) static traffic, where all the connection requests are already known; 2) dynamic traffic, where connection requests arrive in a dynamic fashion and lightpaths are setup on demand. Most previous work focuses on single-fiber networks where each node pair is connected by a single fiber link while recently more and more research is carried out on multiple-fiber networks. The benefit of adding wavelength converters has also been studied. For a recent survey on RWA problem, please see [3] [4].

This paper is an extension of our previously proposed research in [1]. In [1], we propose a RWA algorithm that is restricted to a network with a single fiber per link. In this paper, we extend our algorithm to a network with multi-fibers per link. In addition, we investigate the wavelength converter placement problem under the converter-available assumption.

II. BACKGROUND

Introduced by Frei and Faltings [5], the *Blocking Island* (BI) provides an efficient way of abstracting resources (especially bandwidth) available in a communication network into different levels. In particular, BI clusters parts of the network according to the bandwidth availability. A b-BI for a node x is the set of all nodes of the network that can be reached from x using links with at least b available bandwidth (Fig. 1).

We assume all demands are unicast and the only QoS parameter taken into account is bandwidth. The network physical topology consists of V nodes arbitrarily connected by L bi-directional links. We model it as a network graph $G=(V, L)$. Fig. 1 depicts such a network graph.

A request is defined by a triple: $d_u=(x_u,y_u,\beta_u)$, where x_u and y_u are distinct nodes of the network and β_u is the bandwidth requirement. β-BI has some very useful properties. Below we list a few without proof (for a proof, see [5]).

Unicity: there is one and only one β-BI for a node. Thus if S is the β-BI for a node, S is the β-BI for every node in S.

Partition: β-BI induces a partition of nodes in a network.

[1] This research is supported in part by a grant from the UGC-Hong Kong under the grant AoE/E-01/99.
Corresponding Author email: hamdi@cs.ust.hk

Route existence: give a request $d_u=(x_u,y_u,\beta_u)$, it can be satisfied if and only if the node x_u and y_u are in the same β_u-BI.

Inclusion: If $\beta_i<\beta_j$, the β_I-BI for a node is a subset of the β_j-BI for the same node.

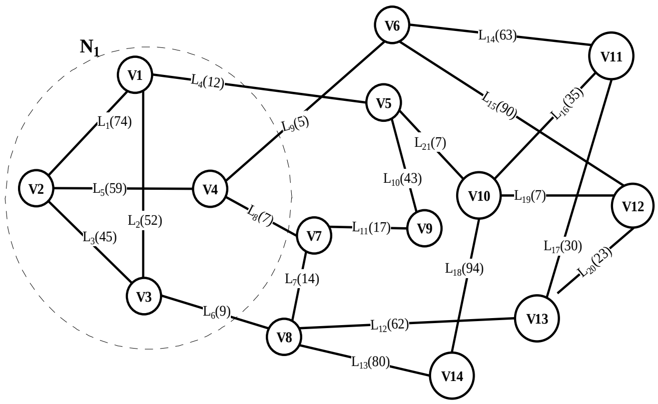

Fig. 1: shows a network topology (NSFNet). N_1={V1, V2, V3, V4} is the 40-blocking island (40-BI) for node V1.

Using the concept β-BI, we can construct a recursive decomposition of Blocking Island Graphs in decreasing order of βs, e.g., $\beta_1>\beta_2>\ldots>\beta_n$. We call this layered structure of Blocking Island Graphs a Blocking Island Hierarchy (BIH). For example according to a demand table and the network topology (Fig. 1), we have such a BIH (Fig. 2).

The most frequent operation in this process is to construct a BIG according to a certain β. It is obtained with a simple greedy algorithm. Starting with an arbitrary node *x*, we add all the nodes which can be reached by links with at least β available bandwidth to form a β-BI. Then starting with another arbitrary node that is not in the previous β-Bis, we repeat the process until all the nodes in the network are included in one of the β-BIs. The complexity of constructing BIG is $O(m)$ [5], where *m* is the number of links in the network.

BI is a natural abstraction of network resources. A β-BIG allows us to get a clear picture about the network load as nodes and links with enough resources are hidden behind an abstract node. In particular, bottlenecks are identified by the interlinks between Blocking Islands.

III. MULTIFIBER BI_RWA ALGORITHM

In this section, we propose a Multifiber Routing and Wavelength Assignment algorithm using BI. The algorithm proposed can be applied to any network with an arbitrary topology.

A. Problem Formulation and BIG Network Model

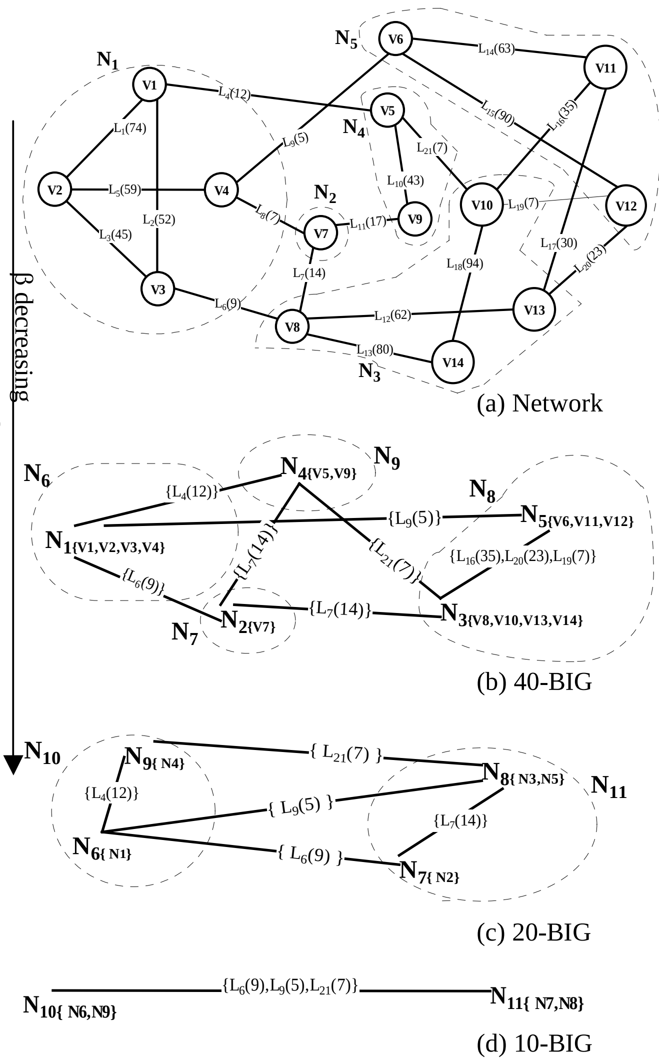

Fig. 2. The blocking island hierarchy for bandwidth requirement {40, 20, 10}. (a) network graph (b) 40-BIG (c) 20-BIG (d) 10-BIG

Define a network topology *G* (*V*, *L*, *W*) for a given WDM optical network, where *V* is the set of nodes, *L* is the set of bidirectional links and *W* is the set of wavelength per fiber link. The set of wavelengths on each fiber link is the same. Each connection request needs to be allocated over a route and assigned one wavelength. The network can be abstracted into $|W|$ blocking island graphs (BIGs). Each BIG starts with one blocking island (BI) representing a wavelength and has the same topology as the original WDM optical network. So the BIG network model BIG(m_1, m_2, …, $m_{|w|}$) can be obtained from a given network topology *G* as follows. The topology of *G* is replicated $|W|$ times denoted by m_1, m_2, …, $m_{|w|}$. Each BIG m_i, which is made of one BI at the beginning, represents a wavelength and the link capacity is 1.

It is obvious that this BIG network model is a simplified blocking island graph. All the properties such as Unicity, Partition and Route Existence still hold. For example, when there is a lightpath (r, λ) where r is the route and λ is the selected wavelength, the *Route Existence property* can be interpreted as whether the route r exists in blocking island λ. For a connection request, instead of routing and assigning a wavelength, we try to find a "best" route in different blocking islands.

B. Proposed Algorithm

Most of the previous work on the RWA problem focuses on single fiber networks. There has been a recent interest in deploying multiple fiber links between node pairs. A multifiber network is an attractive alternative to a network with wavelength conversion capability. An M-fiber W-wavelength network is functionally equivalent to an MW-wavelength network with partial wavelength conversion of degree M [6]. Because of the expensive cost of wavelength converters, multifiber networks may become a viable and economic alternative solution.

Previous research work assumes the same number of fiber links between each node pairs. In our algorithm, we can easily relax this assumption. We simply replace the original BIG with the modified BIG as the initial input Graph. For example, if there are 5 fiber links between AF, AD and BC and there are 3 fiber links between DC, DG, DE and FE. The modified BIG network model is illustrated in Fig. 3. Each fiber link has 3 wavelengths.

We modify the link capacity between each node pairs. The link capacity is equal to the number of fiber links. The rest of the single fiber algorithm can still be applied to the new topology.

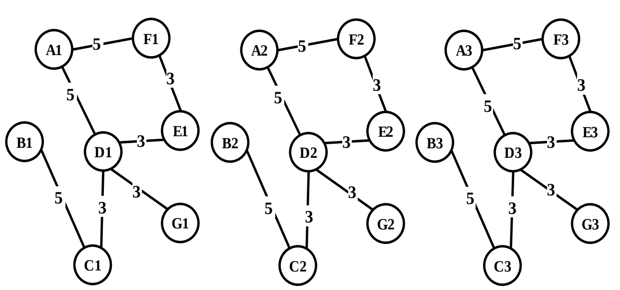

Fig. 3. The BIG network with multiple fibers.

Before describing the algorithm, several concepts need to be defined. The splitting number for a route is equal to the number of BIs that will be newly generated if the route is removed from the current BI. The most loaded link for a route means, in a route, the most wavelengths in this link have been used. Also here we assume traffic is static. Our goal is to maximize the number of accepted requests given a fixed number of wavelengths per fiber link.

Step 1: Transform the network into the BIG model with multiple fibers.
Step 2: Select an unallocated request d, $D = D\text{-}d$. If the request set D is empty then go to step 6.
Step 3: Check the route existence. If all the requests exist, assign the request d to each possible wavelength BIG and calculate K alternate shortest paths. If they don't, go to step 6.
Step 4: Route and wavelength selection.
Now we have a set of candidate routes in different BIGs. Compute the splitting number for each route and the most loaded link for each route. Find one with the minimum splitting number, and the least loaded link among the identified most loaded links.
Step 5: Get the route and corresponding wavelength. Reconstruct the BIGs.
Step 6: If the request set is empty, output the result; otherwise, output "can't be satisfied".

In order to get a more optimal result, a Backtracking Scheme is added to the algorithm as long as time is allowed. In step 3, if not all the requests can be satisfied individually, instead of going to step 8, we backtrack to the previous request and try another of K alternate routes. Notice this algorithm may not find a solution even if one exists, since it looks at K shortest paths only.

If the requests arrive dynamically, we just need to do a few modifications to the original algorithm. For example, we can't order the requests and the backtracking scheme is impossible.

IV. CONVERTER PLACEMENT ALGORITHM

Given a limited number of converters, the optimal placement of converters to reduce the blocking probability is an NP complete problem in an arbitrary mesh network [7]. [8] shows an appropriate placement of limited range wavelength converters could result in reduced blocking probabilities and low distortion of the optical signal. The basic idea is simple: try to find the most congested nodes and put converters on them. Since the BI clustering technique is to balance the load in the whole network by keeping the integrity of the Blocking Island, we could easily decide the bottleneck links using a BIH (Blocking Island Hierarchy).

Static traffic: We assume full conversion at any node. This means there is no wavelength assignment problem. We treat the network as one blocking island with the link capacity equal to the number of wavelengths. Since the traffic is static, we know all the requests in advance. According to the bandwidth requirements, we build the BIH. We then order the requests by decreasing length of their MNH (minimum number of hops) distance and use the lowest level heuristic [5] to do the routing. In figure 1, we can see, the lower level a BI is in the BIH, the smaller it is and thereby we could achieve a computation gain. In addition, the lower a BI is, the more resource is available in the BI. We save the relatively critical links for the future use. This scheme can be viewed as an overall load balancing.

After accommodating all the requests, check the BIH. We could easily pick up the bottleneck links and place converters on those nodes.

For example, Fig. 4(a) is the BIG of the network with full conversion on every node. The bandwidth of each link is 4. After satisfying certain number of connection requests, the bandwidth of each link changes. The final result is in Fig. 4(b). There are two BIs in the 1-BIG: N1 and N2. It is straightforward to see the most congested link is AD and FE, the available bandwidth of both is 0. AD and FE are bottleneck links. If we have two converters, we may put one on A or D and the other one on F or E.

Dynamic traffic: We first need to obtain certain network statistics of the arbitrary network by simulation. Assume full conversion on all nodes. Every time a connection request arrives, reconstruct the 1-BIG (or the 2-BIG, depending on the number of converters you have. For example, in Fig. 4(b), if we construct 2-BIG, we will have 3 BIs and 3 bottleneck links) and record the bottleneck links. After testing enough number of requests, we calculate the tightness of each link. For any link L, we define the number of times it becomes a bottleneck link as B_L and the total number of connection requests as N. Then we define *Tightness of a link* as $L = B_L / N$. Order all the links in decreasing value of tightness and if the tightness of two links is the same, order them randomly. The first link in the list has the highest priority to be put a converter on one of its two nodes. The second link has the second highest priority and so on.

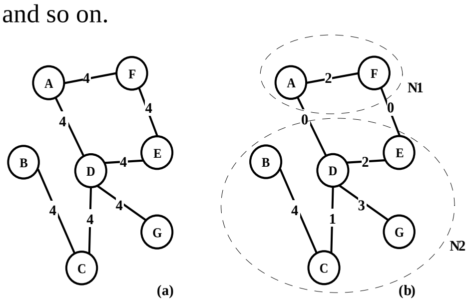

Fig. 4. The placement of converters.

V.

VI. NUMERICAL RESULTS

We use the same Dynamic Traffic Generator model employed in [6]. Calls (requests) arrive at each node according to an independent Poisson process with arrival rate β. An arriving session is equally likely to be delivered to any node in the network. The session holding time is exponentially distributed with mean 1/μ. Thus, the load per *s-d* node pair is $\rho = \beta/N(N\text{-}1)\ \mu$, where N is the number of nodes in the network. Note that a node may engage in multiple sessions and several sessions may be simultaneously conducted between an *s-d* node pair. In our simulation, extensive tests are carried out to ensure a steady state is reached.

We evaluate the performance of the proposed Dynamic BI_RWA algorithm on NSFNet shown in Fig.1, which has 14 nodes and 21 links. The heuristic Dynamic RAW algorithms used in the simulation are fixed routing with first-fit wavelength assignment (FR/FF); fixed routing with most used/pack wavelength assignment (FR/MU); alternate routing with most used/pack wavelength assignment (AR/MU); alternate routing with random wavelength assignment (AR/RAN). The dynamic multi-fiber RWA algorithm using the Blocking Island strategy is called BI_RWA_MultiFiber. The network is treated with even links and unit basic cost. The network with even links means the same number of fibers for every link. The unit basic cost means each fiber for every link has a unit cost. We assume 8 wavelengths per fiber.

First we consider the benefits of using multiple fibers through our BI_RWA algorithms. Fig. 5 shows the call blocking probability of NSFNet with 2 fibers per link and Fig.6 shows call blocking probability of NSFNet with 5 fibers per link. As expected, the blocking performance improves dramatically with the use of multiple fibers. For example, at a blocking probability of 0.04, in the single fiber case (|F|=1, |W|=8), the load is about 38 while in the two fibers case (|F|=2, |W|=8), the load is about 90, the throughput increases by nearly 137%; in the five fibers case (|F|=5, |W|=8), the load is about 235, the throughput increases by more than 500%.

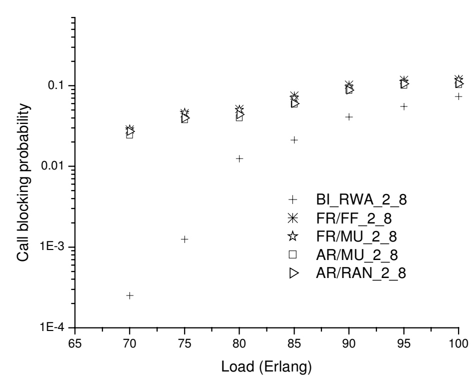

Fig. 5. Blocking probabilities for the NSFNet with 2 fibers per link. BI_RWA_2_8 means BI_RWA algorithm with |F|=2 and |W|=8.

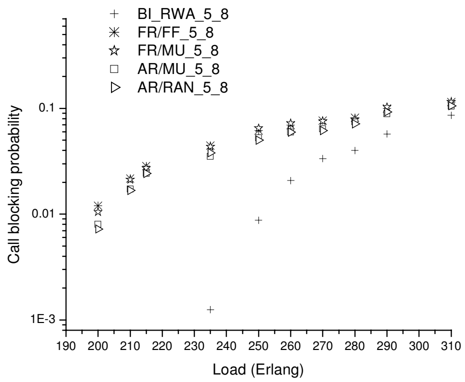

Fig. 6. Blocking probabilities for the NSFNet with 5 fibers per link. E.g. BI_RWA_5_8 means BI_RWA algorithm with |F|=5 and |W|=8.

All these results indicate that when the load is relatively low in each case (that means the RWA algorithm play more important role in handling resources since there are more free resources and with a better management and allocation, a request is more likely to be accepted), the BI_RWA performs much better than the other four algorithms. For example, at a Load = 80 in the 2 fibers case (|F|=2, |W|=8), the call blocking probability of BI_RWA_2_8 is only 2.5×10^{-3}, compared to 5.15×10^{-2} for FR/FF_2_8, 5.05×10^{-2} for FR/MU_2_8, 4×10^{-2} for AR/MU_2_8, and 4.4×10^{-2} for AR/RAN_2_8; at a Load =250 in the 5 fibers case (|F|=5, |W|=8), the call blocking probability of BI_RWA_5_8 is only 8.75×10^{-3}, compared to 6.175×10^{-2} for FR/FF_5_8, 6.5×10^{-2} for FR/MU_5_8, 5.25×10^{-2} for AR/MU_5_8, and 5.1×10^{-2} for AR/RAN_5_8.

We can also see that the performance of the other 4 heuristic algorithms in Fig. 5 and Fig. 6 are almost identical although the FR/MU gives a slightly better result.

VII. CONCLUSION

In this paper, inspired by some artificial intelligence abstraction concepts, we have proposed a new multifiber RWA algorithm called multifiber Blocking Island RWA. In addition, we have used the blocking island abstration to design a converter placement algorithm. All our algorithms are shown to simple, flexible, intelligent and robust. We have also shown that these algorithms can be applied to any network with any topology. Our simulation results have demonstrated that our algorithms outperform state-of-the-art algorithms in this area.

REFERENCES

[1] D. Zhemin and M. Hamdi, "A Simple and Intelligent Routing and Wavelength Assignment Algorithm for All-Optical Networks, " *SPIE Optical Networking and Communicationns Conference* (Opticom), 2001, pp. 210-226.

[2] I. Chlamtac, A. Ganz and G. Karmi, “Lightnet: Lightpath Based Solutions for Wide Bandwidth WANs,” *INFOCOM* ’90, vol. 3, 1014-1021, 1990.

[3] J. Y. Yoo and S. Banerjee, “Design, Analysis, and Implementation of Wavelength-routed All-optical Networks: Routing and Wavelength Assignment Approach,” *IEEE Commun. Survey ’98*. http://www.comsoc.org/pubs/surveys/Yoo/yoo.html.

[4] H. Zang, J.P. Jue, and B. Mukherjee, ``A Review of Routing and Wavelength Assignment Approaches for Wavelength-Routed Optical WDM Networks," *Optical Networks Magazine*, vol. 1, no. 1, Jan. 2000.

[5] C. R. Frei, “Abstraction Techniques for Resource Allocation in Communication Networks,” PhD thesis, Swiss Federal Institute of Technology – Lausanne, 2000.

[6] A. Mokhtar and M. Azizoglu, “Adaptive Wavelength Routing in All-optical Networks,” *IEEE/ACM Trans. Networking*, vol. 6, 197-206, Apr. 1998.

[7] K. C. Lee and V. O. K. Li. “A Wavelength-Convertible Optical Network, “ *IEEE Journal of Lightwave Technology*, 11(5/6): 962-970, May/June 1993.

[8] K. R. Venugopal, M. Shivakumar, P. S. Kumar, “A Heuristic for Placement of Limited Range Wavelength Converters in All-optical Networks, “ *INFOCOM*’99, vol. 2, 908-915, 1999