

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/3235817>

# An efficient algorithm for virtual-wavelength-path routing minimizing average number of hops

Article in IEEE Journal on Selected Areas in Communications · December 2003

DOI: 10.1109/JSAC.2003.818228 · Source: IEEE Xplore

---

CITATIONS

25

---

READS

114

2 authors, including:



[Narayanaswamy Balakrishnan](#)

Indian Institute of Science

209 PUBLICATIONS 2,351 CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:



Computational Electromagnetics [View project](#)



Million Books to the Web [View project](#)

# An Efficient Algorithm for Virtual-Wavelength-Path Routing Minimizing Average Number of Hops

Harsha V. Madhyastha and N. Balakrishnan

**Abstract**—In this paper, we present a novel heuristic algorithm for routing and wavelength assignment in virtual-wavelength-path (VWP) routed wavelength-division multiplexed optical networks. We are the first to take up the approach of both minimizing the network cost, as well as maximizing the resource utilization. Our algorithm not only minimizes the number of wavelengths required for supporting the given traffic demand on any given topology, but also aims to minimize the mean hop length of all the lightpaths which in turn maximizes the resource utilization. The algorithm initially assigns the minimum hop path to each route and then performs efficient rerouting to reduce the number of wavelengths required while also trying to minimize the average hop length. To further reduce the network cost, we also propose a wavelength assignment procedure for VWP routed networks which minimizes the number of wavelength converters required. Our algorithm has been tested on various topologies for different types of traffic demands and has been found to give solutions much better than previous standards for this problem.

**Index Terms**—Network cost, resource utilization, routing and wavelength assignment (RWA), virtual-wavelength-path (VWP) routed network, wavelength conversion, wavelength-division multiplexed (WDM) optical network.

## I. INTRODUCTION

WITH THE rapid growth of the Internet and the ever-increasing demand for voice and video transmission, wavelength-division multiplexed (WDM) optical networks have assumed prime importance. By allowing several channels to be routed on the same fiber on different wavelengths, the capacity of each link is increased tremendously. However, this also calls for more efficient planning before provisioning lightpaths. The problem of assigning routes and wavelengths to lightpaths, called the routing and wavelength assignment (RWA) problem, has been studied widely in literature [1], [2]. As provisioning of an extra wavelength involves considerable increase in network cost, the objective is to minimize the number of wavelengths required, called the network wavelength requirement (NWR). The main constraint on this problem is the fact that the cross-connects at each node are assumed not to have any wavelength conversion capability, which implies that the same wavelength is assigned to a lightpath on all the

links along which it is routed. Such networks are said to be wavelength-path (WP) routed. This constraint can be eased in virtual-wavelength-path (VWP) routed networks which was introduced in [3]. Here, all cross-connects are assumed to have full wavelength conversion capability, i.e., any incoming lightpath can be assigned to any wavelength on the output side.

With the removal of the wavelength-continuity constraint, the problems of routing and wavelength assignment become independent. Now, the NWR becomes equal to the the link wavelength requirement (LWR) of the maximum loaded link since wavelengths to a lightpath can be assigned independently on each link through which it passes. Hence, the problem of minimizing NWR reduces to that of minimizing the maximum LWR. Yet, the routing algorithm itself is NP-Hard and there can exist no deterministic algorithm which gets to the optimal solution always. Though wavelengths can be assigned at random after the routing phase, it is prudent to efficiently allocate the wavelengths so as to minimize the number of converters required because wavelength converters also add to the overall network cost.

Out of the literature already existing in this area, the most efficient RWA algorithm for VWP routed networks has been that proposed by Nagatsu *et al.* [4]. This heuristic initially follows priority-based routing, where the priority is the product of the minimum number of hops between the source and the destination and the number of channels yet to be routed between them. The route assigned is the one with minimum sum of link weights where the weightage of each link is the number of channels already routed through it. In the next phase, rerouting is done to reduce the maximum LWR. A genetic algorithm based on ant-colony optimization was proposed for this problem by Varela and Sinclair [5]. Though it performed admirably well, it did not match up to the standards of [4]. Another scheme was presented, specifically for the COST 239 European Optical Network by Tan and Sinclair [6]. This problem was also tackled using an integer linear programming (ILP) formulation of the problem by Wauters and Demeester [7]. The efficacy of using ILP formulations for solving this problem is very less as the time and space complexity involved are huge and it becomes impractical to use these for large networks with dense traffic. A comprehensive summary of existing approaches for RWA in WP and VWP networks is given in [8].

In our paper, we adopt a new approach toward solving this problem of RWA in VWP routed networks. Apart from minimizing the NWR in order to reduce the network cost, we also take up the objective of maximizing resource utilization. One of the standard metrics for resource utilization is average weighted hop count. This is defined as the average number of physical

Manuscript received January 2, 2003; revised June 23, 2003.

H. V. Madhyastha was with the Department of Computer Science and Engineering, Indian Institute of Technology Madras, Chennai 600036, India. He is now with the Department of Computer Science and Engineering, University of Washington, Seattle, WA 98195 USA (e-mail: harsha@u.washington.edu).

N. Balakrishnan is with the Supercomputer Education and Research Center, Indian Institute of Science, Bangalore 560012, India (e-mail: balki@serc.iisc.ernet.in).

Digital Object Identifier 10.1109/JSAC.2003.818228

hops traversed by one unit of traffic. In our problem setting, the smallest traffic unit is the amount of traffic that can be carried on a single lightpath. Hence, average weighted hop count is equivalent to the average number of physical hops taken up by a lightpath. Based on this point of view, we propose a heuristic algorithm for routing lightpaths to minimize both NWR and the average hop length of a lightpath. Though the problem of wavelength assignment is disjoint from that of routing in VWP routed networks, it is necessary to assign wavelengths so as to minimize the number of wavelength conversions because wavelength converters add to the overall cost of the network. We present a wavelength assignment algorithm for this purpose, which aims to minimize the number of wavelength converters required. Therefore, our routing algorithm when used in combination with the wavelength assignment procedure we propose will achieve the objective of minimizing network cost, as well as maximizing resource utilization.

The remaining part of this paper is organized as follows. In Section II, we explain in detail the heuristic algorithm we propose for VWP routing and present an example in Section III to outline its working. The procedure for wavelength assignment is presented in Section IV. The results of the simulations we conducted to compare the performance of our heuristic with that of [4] are presented in Section V. Finally, we conclude and lay down some directions for future work in this area in Section VI.

## II. ROUTING ALGORITHM

As explained in the previous section, our approach to finding the optimal solution for the RWA problem in VWP routed networks tries to not only minimize the number of wavelengths required [called the network wavelength requirement (NWR)] but also aims to minimize the average hop length. The input required for this problem is the physical topology of the network under consideration and the number of lightpaths required to be established between each pair of nodes in this network. All the links in the network, as well as all lightpaths are assumed to be bidirectional, i.e., when a lightpath from  $S$  to  $D$  is routed through link  $(I, J)$ , then bandwidth on a wavelength is to be reserved not only on link  $(I, J)$ , but also on link  $(J, I)$ . Initially, each lightpath is assigned to the route with minimum number of physical hops between source and destination (which can be determined using Dijkstra's algorithm). Later, rerouting is performed to reduce the maximum LWR, which is also the NWR, while also trying to minimize the average hop length. Assume that the links are stored in decreasing order of LWR in a list called *LINKS* and associated with each link  $L$  is a list called *ROUTES* which stores the lightpaths passing through link  $L$ , in the form  $(source, destination)$ , in increasing order of number of hops. The rerouting procedure is as shown.

- 1) Consider the first link  $L$  in the list *LINKS*.
- 2) Consider the first route  $(S, D)$  in the list *ROUTES* for the link  $L$ .
- 3) Let the route presently assigned from  $S$  to  $D$  be  $N_{i_0} \rightarrow N_{i_1} \rightarrow \dots \rightarrow N_{i_h}$ , where  $S$  is  $N_{i_0}$ ,  $D$  is  $N_{i_h}$  and  $N_{i_x} \rightarrow N_{i_{x+1}}$  forms the link  $L$ .
- 4) Set  $k$  to  $x$ .

- 5) Add all neighbors of node  $N_{i_k}$  except  $N_{i_{k-1}}$  and  $N_{i_{k+1}}$  (either of them may not exist if  $k = 0$  or  $h$ ) to the list *NHBRs*.
- 6) If the list *NHBRs* empty, skip to step 9.
- 7) Consider the first node  $M$  in the list *NHBRs*.
- 8) If the route between  $N_{i_k}$  and  $N_{i_{x+1}}$  is rerouted as  $N_{i_k} \rightarrow M \rightarrow (\text{path } P) \rightarrow N_{i_{x+1}}$ , (where path  $P$  is the path with minimum hops from  $M$  to  $N_{i_{x+1}}$ ), will the load on link  $N_{i_k} \rightarrow M$ , as well as on each of the links along the path  $P$  be lesser than the load on link  $L$ ?
  - a) If yes, change the route from  $S$  to  $D$  as  $N_{i_0} \rightarrow \dots \rightarrow N_{i_k} \rightarrow M \rightarrow (\text{path } P) \rightarrow N_{i_{x+1}} \rightarrow \dots \rightarrow N_{i_h}$ , update load on each link and go back to step 1.
  - b) If no, remove node  $M$  from the list *NHBRs* and go back to step 6.
- 9) if  $k > 0$ , decrement  $k$  and go back to step 5.
- 10) Set  $k$  to  $x + 1$ .
- 11) Add all neighbors of node  $N_{i_k}$  except  $N_{i_{k-1}}$  and  $N_{i_{k+1}}$  (either of them may not exist if  $k = 0$  or  $h$ ) to the list *NHBRs*.
- 12) If the list *NHBRs* is empty, skip to step 15.
- 13) Consider the first node  $M$  in the list *NHBRs*.
- 14) If the route between  $N_{i_x}$  and  $N_{i_k}$  is rerouted as  $N_{i_x} \rightarrow (\text{path } P) \rightarrow M \rightarrow N_{i_k}$ , (where path  $P$  is the path with minimum hops from  $N_{i_x}$  to  $M$ ), will the load on link  $M \rightarrow N_{i_k}$ , as well as on each of the links along the path  $P$  be lesser than the load on link  $L$ ?
  - a) If yes, change the route from  $S$  to  $D$  as  $N_{i_0} \rightarrow \dots \rightarrow N_{i_x} \rightarrow (\text{path } P) \rightarrow M \rightarrow N_{i_k} \rightarrow \dots \rightarrow N_{i_h}$ , update load on each link and go back to step 1.
  - b) If no, remove node  $M$  from the list *NHBRs* and go back to step 12.
- 15) if  $k < h$ , increment  $k$  and go back to step 11.
- 16) Consider the next route  $(S, D)$  in the list *ROUTES* for the link  $L$  and go back to step 3. If there is no route left to be considered, skip to next step.
- 17) Consider the next link  $L$  in the list *LINKS* and go back to step 2. If there is no link left to be considered, then the algorithm terminates.

### A. Remarks on the Algorithm

The essence of the above given algorithm can be summarized as follows. After the initial routing stage, wherein we assign minimum hop path to each lightpath, we try to minimize the load on the link(s) with maximum LWR (say link  $L$  between nodes  $a$  and  $b$ ) by rerouting some lightpath which passes through it. We consider the lightpaths in increasing order of number of hops (taken up by the currently assigned route) because the scope available for rerouting of shorter lightpaths is more (higher number of links are free implies higher degrees of freedom). For this, we partition the set of nodes, through which the lightpath (which is currently under consideration, say  $R$ ) passes, into two subsets—one containing all the nodes occurring before the link  $L$ , i.e., lesser number of hops away from  $a$  than  $b$  (along the route), say set  $A$  and the other containing all

the nodes occurring after the link, i.e., lesser number of hops away from  $b$  than  $a$  (along the route), say set  $B$ .

Now, we consider the nodes in set  $A$  in increasing order of number of hops away from  $a$ , along the route  $R$ . For each node (say node  $N$ ), we enumerate all its neighbors, other than the ones adjacent to it on the current route. These neighbors are considered in random order and for each neighbor (say node  $M$ ), we check whether, if the portion of the lightpath from  $N$  to  $b$  is rerouted through the link between  $N$  and  $M$  followed by the minimum hop path from  $M$  to  $b$ , the load on each of the links through which this rerouted portion passes is lesser than that on link  $L$ . If it is, the route for the lightpath  $R$  is changed as follows. The section of the route from source to  $N$  is retained as before, followed by the link  $N \rightarrow M$ , followed by the minimum hop path from  $M$  to  $b$ , followed by the path from  $b$  to destination as in the original route.

If even after considering all the nodes in set  $A$ , no rerouting was possible, then the similar procedure is repeated with the nodes in set  $B$ . The only difference being that for these nodes, we try to find an alternate path passing through the minimum hop path from one of their neighbors to the node  $a$ . If the lightpath  $R$  could not be rerouted, we move onto the next route (with least number of hops among the remaining routes) through link  $L$  and try rerouting it. If all routes on link  $L$  have been considered, we move onto the next link (the one with maximum LWR among the remaining links). We finally stop when no route on any link can be rerouted. If at any stage rerouting was possible, we start all over again with the least hop path through the maximum loaded link.

The points to be noted in the above rerouting scheme are that first of all, rerouting ensures that the new route does not pass through the link  $L$  ( $a \rightarrow b$ ). So, every rerouting ensures that the traffic on the link under consideration is reduced. The links are considered in decreasing order of LWR, as NWR is the same as the maximum LWR and, hence, reducing NWR requires rerouting of some lightpath passing through the link with maximum LWR. Also, the load on each of the links through which the rerouting is done is lesser than the load that was on link  $L$  before rerouting. This ensures that the algorithm converges to a final solution and terminates in a finite amount of time. However, the other significant point to be noted, the one which helps to minimize the average number of hops, is that whenever a lightpath is rerouted, the number of hops on the lightpath can increase by at most two hops. This will be made clear by Figs. 1 and 2.

In Fig. 1, since Path  $P$  is the minimum hop path from  $M$  to  $b$

$$\begin{aligned} \text{No. of hops on Path } P &\leq \text{No. of hops from } M \text{ to } N \\ &+ \text{No. of hops from } N \text{ to } b \text{ on the current route.} \end{aligned} \quad (1)$$

Since there is a link between  $M$  and  $N$ , this reduces to

$$\begin{aligned} \text{No. of hops on Path } P &\leq 1 \\ &+ \text{No. of hops from } N \text{ to } b \text{ on the current route.} \end{aligned} \quad (2)$$

Now, the increase in the number of hops on the lightpath is given by

$$\begin{aligned} \text{Increase in No. of hops} &= \text{No. of hops on Path } P \\ &+ 1 - \text{No. of hops from } N \text{ to } b \text{ on the current route.} \end{aligned} \quad (3)$$

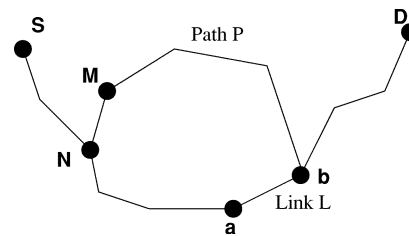


Fig. 1. Rerouting from some node in set  $A$  to  $b$ .

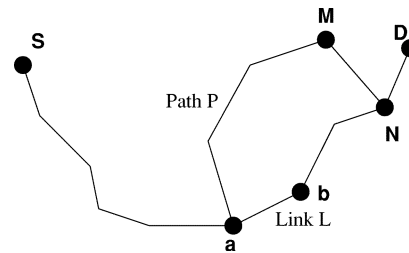


Fig. 2. Rerouting from some node in set  $B$  to  $a$ .

Using (2) and (3)

$$\text{Increase in No. of hops} \leq 2 \quad (4)$$

The proof for the case when rerouting is done from a node in set  $B$  to  $a$  is similar. So, in either case, the maximum increase in number of hops on the lightpath from  $S$  to  $D$  is two. Using this constrained form of rerouting, we not only manage to minimize the maximum LWR (and, hence, the NWR), but also minimize the average number of hops.

In light of the fact that the rerouting we employ ensures that the number of physical hops on the lightpath chosen for rerouting does not increase by more than two, a further optimization can be done to our routing algorithm. In step 5 of our algorithm, we add all the neighbors of node  $N_{i_k}$  to the list  $NHBRs$ . This operation must be carried out such that when the nodes in the list are accessed (node  $M$  in step 7), they are done so in the increasing order of number of hops on the shortest path from  $M$  to  $N_{i_{x+1}}$ , i.e., number of hops on path  $P$ . This ensures that rerouting is always performed along the shortest available path and even though the number of hops cannot increase by more than two, this additional step further increases the probability of rerouting leading to a reduction in the number of hops. Similar considerations must also be taken into account in step 11.

Yet another improvement that can be performed is in step 8 of our algorithm. In case the shortest path  $P$  from  $M$  to  $N_{i_{x+1}}$  passes through some node  $N_{i_y}$ , where  $y \in [x+2, h]$ , then the route from  $S$  to  $D$  can instead be rerouted as  $N_{i_0} \rightarrow \dots \rightarrow N_{i_k} \rightarrow M \rightarrow (\text{path } P) \rightarrow N_{i_y} \rightarrow \dots \rightarrow N_{i_h}$ , where  $P$  is the minimum hop path from  $M$  to  $N_{i_y}$ . Similarly, in step 14, in the case when the shortest path  $P$  from  $N_{i_x}$  to  $M$  passes through some node  $N_{i_y}$ , where  $y \in [0, x-1]$ , then the route from  $S$  to  $D$  can instead be rerouted as  $N_{i_0} \rightarrow \dots \rightarrow N_{i_y} \rightarrow (\text{path } P) \rightarrow M \rightarrow N_{i_k} \rightarrow \dots \rightarrow N_{i_h}$ , where  $P$  is the minimum hop path from  $N_{i_y}$  to  $M$ . Note that the proof given above to demonstrate that our rerouting increases the number of hops by at most two, holds even with this modification.

### B. Complexity Analysis

Since the scenario we are considering is that of planning the network before provisioning lightpaths, the time complexity of the algorithm is not much of an issue. However, we analyze the worst-case time complexity of the algorithm we have proposed, as well as that of the algorithm given in [4] to show that the increase in complexity is not much considering that our algorithm provides an extra feature of minimizing the average number of hops.

Assume that the network has  $V$  nodes and  $E$  links, and the total number of demands is  $T$ . Let us first analyze the complexity of Nagatsu *et al.* heuristic. The initial routing requires the execution of a single-source shortest path algorithm for determining the route of every demand. Hence, the complexity of this phase is  $O(TV^2)$ . Rerouting essentially involves trying to find an alternate route for any lightpath passing through the link with maximum LWR. Finding an alternate route for one lightpath is  $O(V^2)$  and since each lightpath passing through every link with maximum LWR is considered for rerouting, in the worst case, every lightpath maybe tried to be rerouted. So, one successful rerouting takes  $O(TV^2)$ . As the NWR reduces by at least one after every rerouting, the overall complexity is  $O(TV^2 + W_{\max}TV^2)$ , which is equal to  $O(W_{\max}TV^2)$ , where  $W_{\max}$  is the NWR after the initial routing phase.

In our algorithm, the complexity of the initial routing phase is essentially that of finding all-pairs shortest paths which is  $O(EV \log V)$ . Since our rerouting requires only the minimum hop path between every pair of nodes, which are anyway determined in the initial routing phase, the complexity of determining whether a lightpath can be rerouted is  $O(h_{\max}V)$ , where  $h_{\max}$  is the number of hops on the route of the longest lightpath. To reduce the NWR by one, this rerouting might have to be tried for every route on each of its links, which is  $O(h_{\max}^2TV)$ . Since the NWR reduces by at least one in each iteration, the overall complexity of our algorithm is  $O(EV \log V + W_{\max}h_{\max}^2TV)$ , where  $W_{\max}$  is same as above. Since  $h_{\max}$  is of the same order as  $V$ , our algorithm is roughly of one order higher in complexity than that of [4]. This increase in complexity is certainly manageable because as already mentioned our algorithm is employed for preplanning of the network. Also, in addition to minimizing NWR, our algorithm also minimizes the average number of hops.

### C. Survivability

An issue of great significance to be considered in today's networks is that of survivability, i.e., the ability of the network to cope with different kinds of failures. The scenario usually considered in the literature is the single-link failure model, wherein it is assumed that at most one link can fail in the network at any point of time and none of the nodes are prone to failure. Making the network survivable in the face of such failures is of great importance as every lightpath carries traffic of the order of gigabits per second and since each link supports lightpaths on several wavelengths (existing technology supports as many as 320), the failure of a link even for a short period of time can result in huge loss of data. Trying to find alternate routes for traffic after the failure is detected will not suffice as some delay will

be involved in detection, during which huge amounts of traffic will have to be buffered which is not feasible. Hence, for every lightpath, another alternate lightpath (called its secondary) is set up between the same  $(S, D)$  pair which is link-disjoint with the primary lightpath. So, at all times, the same data is sent on both the lightpaths and at the destination, if the power on the primary lightpath is low, failure on the primary lightpath is detected and data is read from the secondary lightpath.

This feature to support survivability can be easily accommodated into our routing algorithm. Initially, as already explained in Section II, each primary lightpath will be assigned the minimum hop path from the source to the destination. Now, in addition, the secondary for every lightpath is assigned the shortest path from source to destination in the network without the links through which the primary lightpath is routed.<sup>1</sup> Also, the set *ROUTES* now comprises of the primary as well as secondary lightpaths assigned to each demand. When a route  $(S, D)$  is considered for rerouting, exactly the same procedure outlined above can be followed except that the links along which the alternative path (the secondary if the primary is being considered or *vice-versa*) from  $S$  to  $D$  is routed are not considered. Specifically, in steps 5 and 11 of our algorithm, only those neighbors  $M$  of  $N_{i_k}$  are added such that neither  $N_{i_k} \rightarrow M$  nor  $M \rightarrow N_{i_k}$  are present in the alternative path. Also, in steps 8 and 14, path  $P$  has to be the minimum hop path without considering the links in the alternative route. This ensures that the routes of the primary and secondary lightpaths always remain link-disjoint. At the termination of the algorithm, the path with lesser number of hops out of the two alternative paths can be chosen as the primary lightpath.

The main drawback with this approach is that the important property of our rerouting procedure that the number of hops never increases by more than 2 does not hold any more. Also, the minimum hop path  $P$  will have to be determined in each iteration, as the links in the alternative path will have to be excluded from consideration, and this will lead to an increase in complexity. However, this clearly shows that our algorithm can be easily modified to work in a survivable network and the modified algorithm can be expected to work just as efficiently.

### III. ILLUSTRATIVE EXAMPLE

We now present an example scenario to illustrate how our algorithm achieves its objective of not only minimizing the NWR but also minimizing the average number of hops. We compare the performance of our algorithm in this example with that of the algorithm proposed in [4]. Consider the ten-node network in Fig. 3, with the demands (1, 7), (4, 2), and (5, 6) to be satisfied. The execution of the algorithm proposed in [4] is shown in Fig. 3. Since there is only one demand between each  $(S, D)$  pair, this algorithm considers the demands in the decreasing order of the minimum hop path from source to destination. Hence, as shown in Fig. 3(a), the demand (1, 7) gets routed first along its minimum hop path because the minimum number of hops for the lightpaths (1, 7), (4, 2), and (5, 6) are 4, 2, and 1, respectively. Next, the demand (4, 2) gets routed, but not along

<sup>1</sup>It is assumed that the network is two-edge connected and so, there are two link-disjoint paths between any pair of nodes in the network.

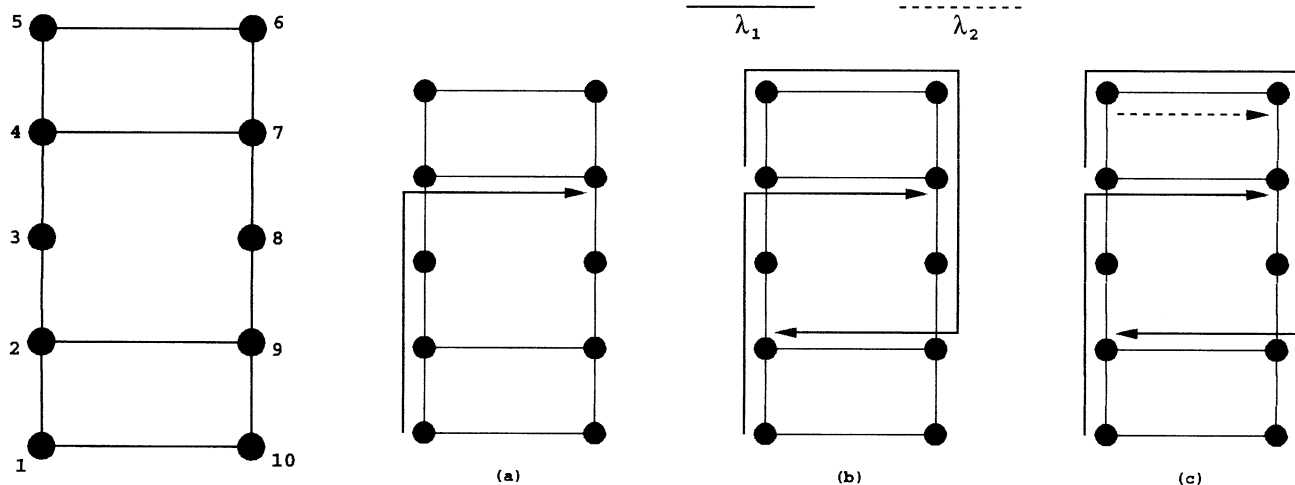


Fig. 3. Ten-node example physical topology and working of Nagatsu *et al.*'s heuristic.

the minimum hop path. Since Nagatsu *et al.*'s algorithm assigns the weight of a link as the number of lightpaths routed through it and each lightpath is routed along the least weight path, the route  $4 \rightarrow 5 \rightarrow 6 \rightarrow 7 \rightarrow 8 \rightarrow 9 \rightarrow 2$  is chosen. Finally, (5, 6) gets routed. Now, the link (5, 6) has the highest LWR of two, which makes the overall NWR also two. The lightpaths (4, 2) and (5, 6) are candidates for rerouting but there is no rerouting possible which will decrease the NWR. Hence, the execution of the algorithm terminates leading to a NWR of two and average hop length equal to  $(4 + 6 + 1)/(3) = 3.67$ .

Now, let us step through the execution of our proposed heuristic algorithm. As explained in Section II, initially each lightpath is assigned to the minimum hop path between the source and destination. The state of the network after this phase is shown in Fig. 4(a). As can be seen from this figure, links (2, 3) and (3, 4) have their LWR equal to two, which leads to a NWR of two. Assume first link (2, 3) is considered, in which the lightpath (4, 2) is first taken up for rerouting as it is the one with lesser number of hops among the two lightpaths routed through this link. Clearly, there does not exist an alternative route for this lightpath such that, after rerouting, the LWR of all the new links through which the route passes will be lesser than two. So, the lightpath (1, 7) is next considered for rerouting. In the first phase of the rerouting, the nodes on the route which are nearer to two than three (set *A* in Section II) are considered. So, node 2 is first considered, whose only neighbor not on the current route is node 9. The shortest path from 9 to 3 passes through 2, and so, rerouting is not possible. Similarly, the only neighbor of node 1 (which is considered next) outside the current route is 10, from where the shortest path to 3 passes through 2 and so, rerouting is ruled out. Next, the nodes on the other side of link (2, 3) are considered (set *B* in Section II). Both the neighbors of node 3 are on the current route and so, node 4 is considered. Its only candidate neighbor is 5, from where the shortest path to 2 passes through 3, and so, rerouting is not possible. Finally, node 7 is considered. Not only does the shortest path from its neighbor node 8 to node 2,  $(2 \rightarrow 9 \rightarrow 8)$ , not pass through any node on the current route but the LWR will be one on all the new links through which the route will pass. So, the lightpath from 1 to 7 will be rerouted from  $(1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 7)$  to

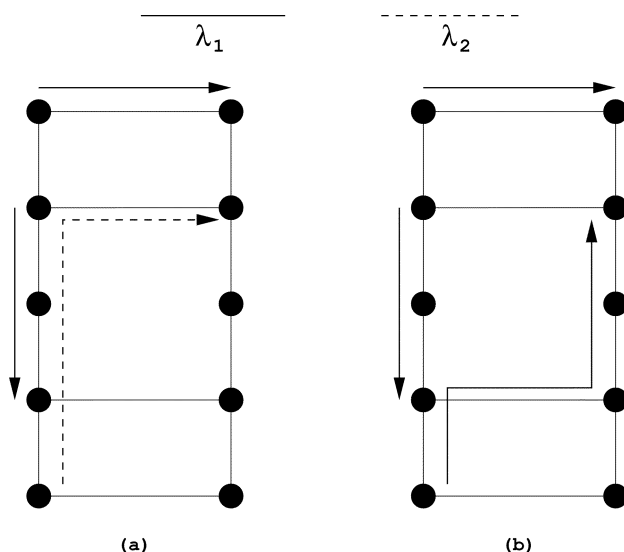


Fig. 4. Working of our routing algorithm.

$(1 \rightarrow 2 \rightarrow 9 \rightarrow 8 \rightarrow 7)$ . The updated state of the network is shown in Fig. 4(b), wherein it can be seen that the NWR is now down to 1, which is the least possible value. So, the algorithm terminates with NWR equal to 1 and the average hop length obtained is  $(2 + 4 + 1)/(3) = 2.33$ .

On the whole, this example clearly showed that our algorithm not only minimizes the NWR (in this case, even better than Nagatsu *et al.*'s heuristic) but also reduces the average hop length of lightpaths significantly. This is due to the extra property of our algorithm that the number of hops on a lightpath increases by at most two in each iteration of the rerouting phase.

#### IV. WAVELENGTH ASSIGNMENT

After performing the routing to minimize the NWR, we need to assign wavelengths for each lightpath on each of the links through which it is routed. In order to reduce the network cost, the wavelength assignment has to be done so as to minimize the number of wavelength converters required. At each node  $N$ , one wavelength converter is required for each lightpath  $R$  which passes through it such that  $R$  has been assigned different

wavelengths on the two links incident at  $N$  through which  $R$  passes. For the wavelength assignment we follow a greedy approach, which is slightly similar to the one followed in [4]. The algorithm given in [4] is meant for WP routed networks and so, the number of wavelengths required is decided by the wavelength assignment procedure. However, in our case, as we are considering VWP routed networks, the number of wavelengths required is determined by the routing procedure itself as NWR is equal to the maximum LWR. The purpose of our wavelength assignment procedure is to assign the lightpaths to the wavelengths in the range 1 to NWR, minimizing the number of wavelength converters. The wavelengths to be assigned can be assumed to be sequentially numbered from 1 to NWR. Also assume that all the lightpaths are arranged in decreasing order of number of hops in the list *ROUTES*. Let the function, assigned ( $W, L$ ), take the value 1 if wavelength  $W$  has been assigned to some lightpath on link  $L$  or else 0.

- 1) Set *WAVE-NUM* to 1.
- 2) Let  $R$  be the first lightpath in the list *ROUTES*.
- 3) Is  $\sum_L$  assigned (*WAVE-NUM*,  $L$ ) = 0 over all the links  $L$  through which lightpath  $R$  passes? If yes, assign wavelength *WAVE-NUM* to the lightpath  $R$  on all its links and remove lightpath  $R$  from the list *ROUTES*.
- 4) If all the lightpaths in the list *ROUTES* have already been considered, skip to next step. Else, let  $R$  be the next lightpath in the list *ROUTES* and go back to step 3.
- 5) If *WAVE-NUM* is not equal to NWR, increment *WAVE-NUM* and go back to step 2.
- 6) Let  $R$  be the first lightpath in the list *ROUTES*.
- 7) Let  $W$  be the serial number of the wavelength which minimizes  $\sum_L$  assigned ( $W, L$ ) over all the links  $L$ , through which lightpath  $R$  passes and on which lightpath  $R$  has not yet been assigned a wavelength. If more than one  $W$  satisfies this property, select the least  $W$  among them.
- 8) Assign wavelength  $W$  to the lightpath  $R$  on all the links  $L$ , through which lightpath  $R$  passes and on which lightpath  $R$  has not yet been assigned a wavelength and where assigned ( $W, L$ ) = 0.
- 9) If lightpath  $R$  has not been assigned a wavelength on all the links through which it passes, go back to step 7. Else, remove lightpath  $R$  from the list *ROUTES*.
- 10) If the list *ROUTES* is empty, then the algorithm terminates. Else, go back to step 6.

The synoptic explanation of the above algorithm is as follows. First, we take up each wavelength sequentially, trying to assign lightpaths to that wavelength in priority order (path with higher number of hops has higher priority) such that the same wavelength can be assigned to that lightpath on all the links through which it passes. When no lightpath can be assigned to a particular wavelength, we move on to the next wavelength and start assigning lightpaths to it. When all the wavelengths (from 1 to NWR) have been considered, we move onto the next phase. Here, we consider the lightpaths in decreasing order of number of hops. For each lightpath  $R$ , we determine the wavelength  $W$  which is as of now unassigned on the maximum number of links (compared with other wavelengths), through which the lightpath

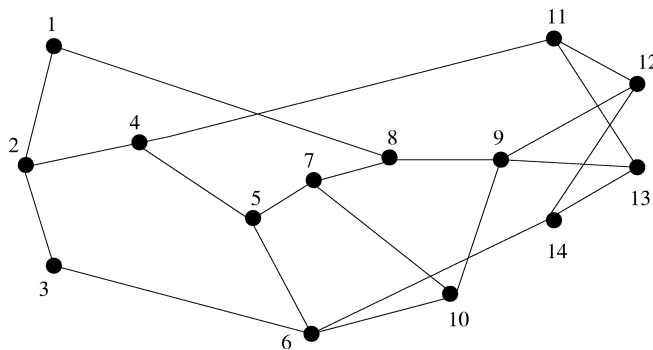


Fig. 5. Topology of NSFNET.

$R$  passes and on which the lightpath  $R$  has not been assigned a wavelength yet. The lightpath  $R$  is assigned to wavelength  $W$  on all the links where it has not been assigned a wavelength yet and on which wavelength  $W$  is unassigned. We repeat the previous two steps until lightpath  $R$  has been assigned to some wavelength on each of the links through which it passes. Then, we move on to the next lightpath (the one with maximum number of hops among the remaining lightpaths) and repeat the process. Throughout this procedure, we only consider wavelengths in the range 1 to NWR.

## V. RESULTS

To determine the optimality of our algorithm, we tested it on networks of various physical topologies with a wide variety of traffic distributions. We compared our results with that obtained by using the heuristic proposed in [4], the one currently considered to be the best for this problem. Owing to the extra constraints in our rerouting procedure, our heuristic certainly took more number of iterations in the rerouting phase to get to the final solution. However, as our problem setting involves static planning of the network before provisioning lightpaths, the time of execution is not much of an issue and as discussed in Section II-B, in this context, the increase in complexity is not much. In fact, other works on this problem such as [5] and [6] propose genetic algorithms which have significantly higher complexity and runtimes of the order of hours. In our case, the time of execution of our heuristic was more or less comparable to that of [4], with both of them taking only a few seconds even on the real-world physical topologies and traffic demands we considered. Though the final solution, in terms of NWR, given by both heuristics was more or less the same in most cases, the average hop length was considerably lesser with our heuristic in almost all cases considered.

Here, we give the results of testing on three standard networks along with their corresponding measured traffic demands as given in the literature. The first network considered is the Pan-European Optical Network (given in [9]), which has 19 nodes and 39 links, with the traffic demand as given in [10]. The second network considered is the NSFNET (shown in Fig. 5), which has 14 nodes and 21 links, with the measured traffic demand taken from [11]. The topology considered for the last network is formed from the 11 central nodes of the

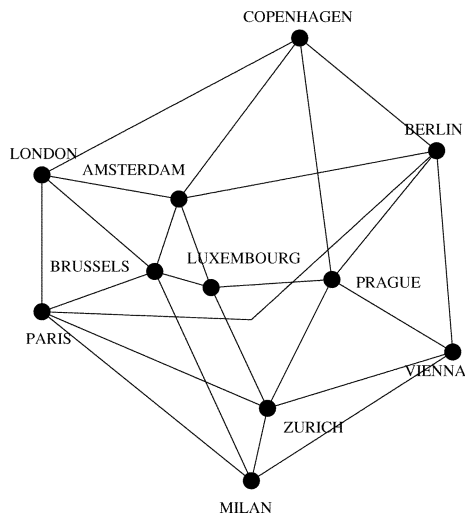


Fig. 6. Topology of European Optical Network.

TABLE I  
RESULTS OBTAINED

S.No.	Nagatsu <i>et al.</i>		Min-Hops	
	NWR	Total No. of hops	NWR	Total No. of hops
1	67	534	69	440
2	26	138	25	113
3	6	101	6	85

European Optical Network (shown in Fig. 6), which has 24 links. The traffic distribution for this network was taken from [6]. The results obtained by executing both our heuristic (called the Min-Hops heuristic), as well as the heuristic proposed in [4] are given in Table I.

As can be seen from the results, our heuristic not only performed as well as the heuristic in [4] in terms of minimizing NWR, but also did much better in minimizing the number of hops. This is due to the fact that our heuristic starts off by assigning each lightpath to the minimum hop path and then performs efficient rerouting such that the number of hops on any lightpath can increase by at most two in a single iteration. As the same set of lightpaths were setup by both heuristics, the total number of hops is an equivalent measure of the average number of hops, which is the average weighted hop count. Thus, the results substantiate our claim that our heuristic achieves the combined objective of minimizing network cost, as well as maximizing resource utilization.

## VI. CONCLUSION

We considered the problem of routing and wavelength assignment (RWA) in virtual-wavelength-path (VWP) routed networks and took up the novel approach of not only minimizing the network cost, in terms of number of wavelengths and number of wavelength converters, but also maximizing the resource utilization, measured by the average weighted hop count. We proposed a heuristic algorithm for routing which not only tries to minimize the number of wavelengths required (NWR) but

also minimizes the average number of hops taken up by a lightpath. We also presented a wavelength assignment procedure which minimizes the number of wavelength converters required. We compared our algorithm with one of the standard algorithms for this problem [4] and found the results to be highly encouraging.

We are currently working on how to tackle the same problem bringing the issue of survivability into consideration, without losing out on the important property of our rerouting procedure that the number of hops increases by at most two. Also, instead of using average weighted hop count as the measure for resource utilization, other standard measures are being considered, which will put forward the need for different heuristics.

## REFERENCES

- [1] R. Ramaswami and K. N. Sivarajan, "Routing and wavelength assignment in all-optical networks," *IEEE/ACM Trans. Networking*, vol. 3, no. 5, pp. 489–500, Oct. 1995.
- [2] D. Banerjee and B. Mukherjee, "A practical approach for routing and wavelength assignment in large wavelength-routed optical networks," *IEEE J. Select. Areas Commun.*, vol. 14, pp. 903–908, June 1996.
- [3] K.-I. Sato, S. Okamoto, and H. Hadama, "Network performance and integrity enhancement with optical path layer technologies," *IEEE J. Select. Areas Commun.*, vol. 12, pp. 159–170, Jan. 1994.
- [4] N. Nagatsu, Y. Hamazumi, and K. I. Sato, "Number of wavelengths required for constructing large-scale optical path networks," *Electron. Commun. Jpn., Part I—Commun.*, vol. 78, no. 9, 1995.
- [5] N. Varela and M. Sinclair, "Ant colony optimization for virtual-wavelength-path routing and wavelength allocation," in *Proc. Congress Evolutionary Computation (CEC'99)*, Washington, DC, 1999, pp. 1809–1816.
- [6] L. G. Tan and M. C. Sinclair, "Wavelength assignment between the central nodes of the cost 239 european optical network," in *Proc. 11th U.K. Performance Engineering Workshop*, Liverpool, U.K., Sept. 1995, pp. 235–247.
- [7] N. Wauters and P. Demeester, "Design of the optical path layer in multi-wavelength cross-connected networks," *IEEE J. Select. Areas Commun.*, vol. 14, pp. 881–892, June 1996.
- [8] R. Ramaswami and K. N. Sivarajan, *Optical Networks: A Practical Perspective*. San Mateo, CA: Morgan Kaufmann, 1998.
- [9] M. Tornatore, G. Maier, and A. Pattavina, "WDM network optimization by ILP based on source formulation," *Proc. IEEE INFOCOM'02*, 2002.
- [10] A. Fumagalli, I. Cerutti, M. Tacca, F. Masetti, R. Jagannathan, and S. Alagar, "Survivable networks based on optimal routing and WDM self-healing rings," *Proc. IEEE INFOCOM'99*, pp. 726–733, Mar. 1999.
- [11] Y. Miyao and H. Saito, "Optimal design and evaluation of survivable WDM transport networks," *IEEE J. Select. Areas Commun.*, vol. 16, pp. 1190–1198, Sept. 1998.



**Harsha V. Madhyastha** received the B.Tech. degree in computer science and engineering from the Indian Institute of Technology, Madras, India, in 2003. He is currently working toward the M.S. degree in computer science and engineering at the University of Washington, Seattle.

His current research interests include traffic grooming in WDM optical networks, study of QoS and fairness issues in telecommunication networks, and information extraction.

Mr. Madhyastha was selected by the Government of India to represent the nation at the Science Research Program, Singapore, in 1998. He was a recipient of a silver medal at the 39th International Physics Olympiad, Padua, Italy, in 1999.





**N. Balakrishnan** received the B.E. degree (honors) in electronics and communication from the University of Madras, Madras, India, in 1972 and the Ph.D. degree in engineering from the Indian Institute of Science, Bangalore, in 1979.

He then joined the Department of Aerospace Engineering, Indian Institute of Science as an Assistant Professor. He is currently a Professor in the Supercomputer Education and Research Centre and the Department of Aerospace Engineering, Indian Institute of Science. He also heads the Division of Information

Sciences. He has over 150 publications in the areas of computational electromagnetics, polarimetric radars, networking and information security, and speech processing.

Dr. Balakrishnan is the Fellow of all Academies of Science and Engineering in India. His work is well recognized through several awards, the most notable being the Padmashree by the President of India, the Millennium Medal of Indian Science Congress, The Alumni Award of Excellence in Engineering Research of IISc and the Vikram Sarabhai Research Award. He is an Honorary Professor of the National Institute for Advanced Studies and the Jawaharlal Nehru Centre for Advanced Scientific Research. He is Director of Bharath Electronics. He is also a Member of the Scientific Advisory Committee to the Cabinet.