In Proceedings of the 2nd IEEE International conference on COMmunication System softWAre and MiddlewaRE (COMSWARE 2007), Bangalore, India, 7-12 January 2007.

Survivable Routing in WDM Weighted Networks

Debasis Mandal[1], **Satyajit Nath**[2], **Bivas Mitra**[3]
Infosys Technologies Ltd, Pune[1], CMC Ltd., Kolkata[2], IIT, Kharagpur[3]
Email: debasis.mandal@gmail.com[1], satyajit.nath@cmcltd.com[2], bivas_mitra@yahoo.com[3]

Abstract: **In this paper, we investigate the problem of routing lightpaths on an arbitrary physical topology following a Design Protection approach, such that virtual topology remains connected even after the failure of a single fiber link. This is called survivable routing. It is known to be an NP-complete problem. To address the problem, first, we have proved that embedded Hamiltonian circuit in a mesh network implies the survivability of its virtual topology. Then using a polynomial time algorithm for generating Hamiltonian circuit i.e. embedded ring virtual topology, we establish lightpaths to the ring. Finally, we design two RWA algorithms to assign lightpaths to other requests in the network, giving priority to wavelength and traffic respectively. We analyze the numerical results obtained for random undirected networks with random normal traffic demands with performance metrics such as maximum one-hop and minimum multi-hop protected traffic, wavelength utilization, number of multi-hops, Buffer size etc.**

Keywords—Physical topology, lightpath, virtual topology, WDM, survivable routing, hamiltonian circuit, Routing and Wavelength Assignment (RWA).

I. INTRODUCTION

Survivability of a network refers to the network's capability to provide continuous service in the presence of failure. A network failure may be mainly due to link or node failure. Since most modern node devices have built-in redundancy that greatly improves their reliability, failure of fiber links is more of a concern [2] as they pass through different atmospheric conditions (like, under oceans). Again, since protection at electronic layer (ATM, IP) is more time-consuming, optical layer provides resource and time effective fault-tolerance even to upper unprotected layers [1]. So we concentrate on survivability to a single fiber link failure (predominant form of failure [1]) through optical layer protection in this paper.

In WDM terminology, *Physical Topology* is a set of nodes interconnected with the pair of fiber links while *Virtual Topology* at the optical layer consists of a subset of the nodes at physical layer interconnected with *lightpaths*. The assignment of free channels of Physical topology to the links in the logical topology is performed by the *Design Algorithm*. And providing survivability to the physical network through virtual topology is called *Design Protection* [1]. In a wavelength-routed WDM network, each fiber link can carry several lightpaths and the failure of even a single fiber can lead to the failure of multiple lightpaths in the virtual topology. This in turn can disconnect the virtual topology and prohibits the optical layer routing. Design protection has come into the picture to provide a protected network at the design phase itself before actually deployment of the network, such that the virtual topology never gets disconnected under the fiber-link failure. This problem is called *Survivable Routing* of a logical topology on the physical topology. In this paper, we address this problem such that the virtual topology remains connected even after the failure of a single fiber link. The survivability routing problem is decomposed into four sub-problems [4][5]:

- *Survivable Topology design* – determines the survivable virtual topology to be imposed on the physical topology based on the traffic demands.
- *Virtual Topology Routing* – computes a physical path for each logical link in the virtual topology.
- *Wavelength Assignment* – deals with assigning a free wavelength along the computed physical path corresponding to each virtual link in the virtual topology.
- *Traffic Routing* – computes a virtual path to route traffic between source and destination nodes in the virtual topology.

To address the first sub-problem, we design a ring virtual topology using Hamiltonian circuit which guarantees survivability. Then we design two RWA (Routing and Wavelength Assignment) algorithms optimizing one-hop traffic to solve the second and third sub-problems. Finally, we route the multi-hop traffics on the virtual topology generated by RWA algorithms, and this addresses the fourth sub-problem.

Given Physical Topology and traffic pattern in the network, design of a survivable virtual topology is NP-hard [3]. Also, survivability against single link failure (*survivable RWA*) in physical topology, introduced by Modiano et al., is proved to be NP-complete [6][7]. Modiano also presented the necessary and sufficient conditions for survivable routing in a network and provided its ILP formulation. Survivability against Shared Risk Link Groups [SRLG], subject to single fiber failure is studied in [5]. In [8], the survivable routing problem for rings is addressed through Euler Paths. Again, [9][10] have used addition of virtual links to make the virtual ring survivable.

The rest of the paper is organized as follows. Section II provides mathematical formulation of the survivability problem and proves that embedded Hamiltonian circuit in a mesh network implies the survivability of its virtual topology. In section III, we propose an algorithm to establish lightpaths using a single wavelength that will take care of survivability of the virtual topology. Section IV provides two polynomial RWA algorithms based on Priority on Wavelength (POW) and Priority on Traffic (POT) to establish additional lightpaths using available wavelengths in each link to optimize other performance criteria. Section V analyzes the complexity of the algorithms presented throughout this paper. Finally, Section VI presents the performance evaluation of the algorithms on random graphs based on various performance parameters via simulation.

Traffic Model

Another goal of this paper is to distribute the traffic uniformly across the network as far as possible. For this purpose and to make the traffic model more realistic, we have used Normal Distribution as statistical measure to measure random traffic generated at each node of WDM network. The standard deviation (σ) and mean (μ) of the distribution are considered constant and provided with predefined values. Normal distribution is a continuous distribution and is defined by the density function,

$$F(x) = (1/\sigma\sqrt{2\pi}) \times e^{-z}, \text{ where } z = (1/2) \times \{(x-\mu)/\sigma\}^2,$$

where, x represents a random variable. First, we have generated an $N \times N$ ($|V|=N$) matrix R of normal random numbers using *Boxmuller program* [11]. Then an $N \times N$ matrix A is generated, which is consequently multiplied by R (modulo 100) to get a uniform picture of normal traffic distribution across all nodes. This generates final normal traffic matrix T, where $(i, j)^{th}$ element t_{ij} represents the long term average traffic demand from node i to node j. The traffic is assumed to be static, i.e. the set of requests is known in *priori* [2]. We used normalized traffic matrix to make our simulation more realistic. Though the average traffic demand may be asymmetric, we generate a symmetric traffic matrix where $t_{ij}= \max(t_{ij}, t_{ji})$. Also we assume diagonal elements (i = j) to be zero, considering no extra traffic sinks or generates at nodes.

II. MATHEMATICAL FORMULATION

A WDM network (physical topology) is modeled by an undirected connected graph $G_p = (V_p, E_p)$, where V_p is the set of N nodes (combination of access station and OXC) and E_P is the set of L fiber optic links ($L \geq N-1$). Each fiber can carry W wavelengths $\Lambda = \{\lambda_1, \lambda_2, \ldots, \lambda_W\}$. We have also used a weighted function $W(e_{ij}) = w_{ij}$, where $w_{ij} \in R^+$, for each edge $e_{ij} \in E_P$. We have to design a survivable virtual topology $G_v = (V_v, E_v)$, where $V_v \subseteq V_p$ and $E_v = \{(x, y)$: $x, y \in V_v$, such that there exists an all-optical lightpath between nodes x and y$\}$; which remains connected even after the failure of single fiber link in G_p. As each lightpath can carry the traffic in both directions, each edge of the virtual topology is bi-directional.

Providing survivability to the network through virtual topology is called *Design Protection*. This problem is addressed as Survivable Virtual Topology Routing [SVTR] in [5]. A lightpath $l_{s,t}$ (or, $l_{t,s}$) in virtual topology G_v is routed through several fiber optic links of the physical topology G_p. We call a routing *survivable* if failure of any fiber link leaves the virtual topology connected, which, in turn, is not possible unless the underlying virtual topology is 2-connected. Again, the virtual topology is 2-connected if the removal of any lightpath does not cause the topology to be disconnected. [6] proves that the above is valid if and only if no two lightpaths of the virtual topology share the same fiber link of the physical topology. But this requires the availability of wavelength converters at nodes to design the virtual topology to compensate the large number of links and wavelengths needed to fulfill the condition. In turn, this greatly enhances the opto-electronic bottleneck i.e. cost of the network resources, buffer-size and processing delays at each node. In this paper, we have implemented the above criteria only to make virtual topology survivable, but without wavelength conversion. Thus we have minimized the number of wavelength converters to a great extent by first optimizing one-hop traffic (to reduce the opto-electronic conversion at each node) for large traffic requests. Multi-hops i.e. the access of SONET layer (for wavelength conversion) are only allowed for small traffics, not assigned by one-hop.

For this purpose, we consider a special type of protected virtual topology called Ring topology due to its embedded DAPs (Disjoint Alternate Paths) and thus 2-connectivity. A Hamiltonian circuit H_c is a special type of ring topology where $V_v=V_p$ and each node excepting n_1 is visited exactly once, i.e., $H_c= n_1e_{12}n_2\ldots n_{n-1}e_{n-1,1}n_1$. We'll assume that Traffic request set is a *complete graph* (i.e. $|E|=C(N,2)$), to prove the lemmas, as it takes care of all sorts of possible requests. Also, we assume no wavelength conversion capability at nodes to prove the lemmas.

The following lemmas provide a sufficient condition for the existence of at least one survivable virtual topology of a physical topology G_p (see also Appendix – I [6] for the background of the lemmas in terms of cut-set).

Lemma 1

A virtual topology $G_v(V_v, E_v)$ on a physical topology G_p is survivable if G_v has *at least* one Hamiltonian circuit.

Proof: The presence of Hamiltonian circuit (Ring) H_c in the virtual topology G_v itself implies the 2-connectivity of virtual topology and $V_v=V_p$ takes care of the recovery after any link failure in G_p. So the underlying virtual topology G_v is survivable.

Lemma 2

For an arbitrary physical topology $G_p(V_p, E_p)$, there exists *at least* one survivable virtual topology $G_v(V_v, E_v)$ on G_p, if G_p contains a Hamiltonian circuit.

Proof: To prove the above Lemma, assume that the physical topology G_p contains a Hamiltonian circuit H_c= $n_1e_{12}n_2\ldots n_{n-1}e_{n-1,1}n_1$. Then we can assign a lightpath in each edge e_{ij} of H_c using a single wavelength λ_1. This generates a virtual topology with the edges of the Hamiltonian circuit. After this, we can add additional links to the virtual topology by assigning the other requests in the physical topology with different wavelengths. So from lemma 1, the underlying virtual topology is survivable.

III. DESIGN OF A SURVIVABLE VIRTUAL TOPOLOGY

The Lemmas in section II can be utilized to develop an algorithm for finding *survivable RWA*, if exists, in G_p. However, from lemma 2 we can infer that if there is a Hamiltonian circuit in G_p, it's possible to get a survivable virtual topology for it. In this section, we'll use a polynomial algorithm for finding Hamiltonian circuit in an arbitrary graph and then assign wavelength to it.

A. *Design of a Hamiltonian Circuit*

To find a Hamiltonian circuit in an arbitrary graph is a NP-complete problem. To address the problem, we have used the polynomial ($O(n^5)$) algorithm of Ashoy [12] which runs on a un-weighted graph. To get a Hamiltonian circuit with maximum traffic, we have applied a cost factor on each edge before running it with some modification. It provides significantly good result for most random weighted undirected networks with random and normalized traffic. Here we'll prove a lemma which will be used to get the Hamiltonian circuit with maximum weight in the weighted topology.

Lemma 3

If a Hamiltonian path is found for an arbitrary weighted topology by *Farthest-neighbor* method, starting with a node n_0, it'll be the Hamiltonian path with maximum cost starting with the node.

Proof: The proof is straightforward. Suppose we have not got the path with maximum weight. This implies that some edge in the path must have been selected, which is not of the maximum weight attached to a node. This contradicts the farthest-neighbor heuristic and so not possible. Again, since we need to visit every node in the graph, it's not possible to leave out any intermediate edge from being considered for selection.

Corollary: Lemma 3 clearly implies that for any graph, if we get a Hamiltonian Circuit by Farthest-neighbor method, starting with each node; the maximum of them will be the overall Maxima.

Now we will design a polynomial algorithm to find a Hamiltonian circuit with maximum weight, if exists in the physical topology.

Algorithm 1

Input is the physical topology G_p with N vertices.

Step 1: Run the *Farthest-neighbor* heuristic (to get the maximum Hamiltonian circuit, if found) for each node, starting with node 1. If Hamiltonian circuits are found, take the maximum circuit and repeat step 1 for next node.

Step 2: If Hamiltonian circuit is not found in Step 1, apply Ashoy algorithm for the node. It'll give multiple Hamiltonian circuits, if exist, considering G_p as only a undirected graph. Take the maximum weighted circuit.

Step 3: After repeating above steps for each node, take the maximum Hamiltonian circuit from them.

B. *Design of a Ring Virtual Topology*

A ring virtual topology $R_v = (V_v, E_v)$ is an ordered set of nodes $(n_1, \ldots, n_p)$, where edge $(n_i,n_{i+1})\in E_v$ for $0<i<p$ and $(n_l,n_p)\in E_v$. To design a virtual topology by wavelength assignment on the routed paths on physical topology, we use a W×L matrix called *link-state matrix* U for the physical topology $G_p(V_p, E_p)$. The state of a link $i\in E$ can be specified by the column vector $\sigma_i= (\sigma_i(1), \sigma_i(2), \ldots, (\sigma_i(W))^T$, where $\sigma_i(j)=1$ if wavelength λ_j is allocated in the link i by some lightpath and $\sigma_i(j)= 0$, otherwise. So the state of the network is given by the matrix $U= (\sigma_1, \sigma_2, ..., \sigma_L)$ [13]. Initially, all wavelengths are available in each link. So we initialize the U matrix by 0. U is updated when a link is assigned by some wavelength, to follow Distinct Wavelength Constraint (DWC). When all the rows of a column in U become 0, we delete the edge from the physical topology Gp.

Apart from Traffic matrix T, an N×N matrix AT (*Assigned Traffic*) is also maintained which stores the information of the lightpaths already established. The requests for which the lightpaths cannot be established after one-hop assignment are stored in another N×N matrix BT (*Blocked Traffic*). The following algorithm will assign a wavelength to all the edges of the Hamiltonian circuit to get a ring Virtual topology.

Algorithm 2

Inputs are the Physical topology G_p and maximum H_c obtained in Algorithm 1.

Step 1: if Algorithm 1 provides a Hamiltonian circuit, create a set of physical links, *Ham*, containing the edges of the Hamiltonian circuit. For all $(u,v)\in$ *Ham*, perform Step 2 and Step 3.

Step 2: Establish a lightpath between the node pair (u,v) using the wavelength λ_1 through the link $(u,v)\in E_p$. Update matrix U by assigning σ_{pk} (i)=1, for all k=1,2…..t. Store the information in $AT(V, E_v)$ matrix by assigning AT[s,d]= T[s,d].

IV. LIGHTPATH ASSIGNMENT TO OTHER TRAFFICS

Without any wavelength conversion, each request can be routed through the above generated ring virtual topology. But this will account for massive traffic in the ring, which each fiber may not support. Also, it'll cause traffic congestion in the network along this path, leaving all other edges free. To avoid this situation, we'll assign different lightpaths (routes) to request (u,v), where (u,v) is not in *Ham,* and use the ring only for backup. This also helps to create a uniform distribution of traffic.

To assign lightpaths to other traffic requests, first update the elements of traffic matrix T by assigning $t_{u,v} = t_{v,u} = 0$ (for symmetry), to indicate that these traffics have been assigned. Then sort the traffic matrix T in descending order. Let *Req* be the ordered set of all (s,d) pairs such that $Req = \{(i_1,j_1), (i_2,j_2), \ldots, (i_n,j_n)\mathbf{:}\ T[i_k,j_k] \geq T[i_{k+1},j_{k+1}],\ 1 \leq i_k,j_k \leq n,\ 1 \leq k \leq n-1\}$. Connection requests will be generated according to this ordered set *Req*, to maximize one-hop traffic. When a connection request arrives, RWA algorithm searches for a physical route $P=<i_1, i_2, \ldots, i_t>$ between (s,d) of the request, such that $\sigma_{ik}(j)=0$ for all k=1, 2, …, t and some j, to satisfy the Wavelength Continuity Constraint (WCC). In brief, by *wavelength availability* we mean, we have to find a row j in U which has 0 entries in each column of the links for the path. The search order is fixed in *priori*, i.e., $\lambda_1, \lambda_2, \ldots, \lambda_W$ in Λ [13].

Now we design two RWA algorithms to find routes and assign lightpaths to other requests based on the Ring virtual topology.

A. Priority on Traffic (POT)

In this RWA algorithm, we find a route for a request and assign it with the first available wavelength from Λ. The algorithm is run for each request from *Req*.

Algorithm 3

Inputs are G_p, *Req,* Λ and maximum H_c.

Step 1: For the connection request (s,d) from *Req*, find the *Shortest Available Path* (SAP) between (s,d) in the physical topology G_p. Let the set of links in the shortest path be $P = <p_1,p_2,\ldots,p_t>$, where $P \subseteq E_p$. Then assign P with a wavelength λ_i from Λ. If λ_i can't be assigned to whole of the path P, try with λ_{i+1} from Λ for the same request, and so on. Thus P is assigned with the first *available* λ_i from Λ.

If there is no available λ_i, try for the next request from *Req*.

Step 2: If λ_i is found, update the link-state matrix U by assigning σ_{pk} (i)=1, for all k=1,2,…,t. Save the updated information in AT matrix by assigning AT[s,d]= T[s,d].

Otherwise, the request is blocked and store this information in the blocked matrix BT by assigning BT[s.d]=T[s,d]. Repeat the algorithm for the next request.

B. Priority on Wavelength (POW)

Here we first try to assign routes of all the requests with wavelength λ_1. Then the requests not satisfied by λ_1 are tried with wavelength λ_2 and so on. The algorithm is run for each λ_i from Λ.

Algorithm 4

Inputs are G_p, *Req,* Λ and maximum H_c.

Step 1: For the wavelength λ_i and for the connection request (s,d) from *Req*, find the *Shortest Available Path* between (s,d) in G_p. Let the set of links in the shortest path is $P = <p_1,p_2\ldots..p_t>$. $P \subseteq E_p$. If P can be assigned with λ_i, according to link-state matrix constraint, go to step 2.

Otherwise, try for next request from *Req*.

Step 2: If λ_i is assigned, update matrix U by assigning σ_{pk} (i)=1, for all k=1,2,…,t. Store the information in AT matrix by assigning AT[s,d]= T[s,d].

Otherwise, the request is blocked and store this information in BT by assigning BT[s.d]=T[s,d]. Repeat the algorithm for the next wavelength λ_{i+1}.

C. Multi-hops wavelength Assignment

Now we assign multiple wavelengths to a lightpath for the requests not assigned by one-hop (POW or POT). This algorithm accounts for all requests to be satisfied.

Algorithm 5

Inputs are G_V, *Req* and BT.

Take a request from *Req* corresponding to BT and route it on the virtual topology G_V. The numbers of wavelengths it get across its path are the number of multi-hops. Repeat this for each request in Blocked Traffic matrix BT.

V. COMPLEXITY ANALYSIS OF ALGORITHMS

Now we analyze the complexities of the algorithms in this paper and prove that the RWA heuristics we have designed run in polynomial time. Suppose, the physical topology has 'n' number of nodes.

Algorithm 1: For each node, the Farthest-neighbor heuristic has $O(n^2)$ complexity and the Ashoy algorithm has complexity $O(n^4)$. The algorithm runs for each node. So the overall complexity is $n\times[O(n^2)+ O(n^4)] \approx O(n^5)$.

Algorithm 2: Creating the set *Ham* of all nodes takes $O(c_1\times n)$ time and assigning lightpath for 'n' edges takes $O(c_2\times n)$ time. So the overall complexity of the algorithm is $[O(c_1n)+O(c_2n)] \approx O(n)$.

Algorithm 3 and 4: Finding shortest path for each request has $O(n^2)$ complexity. It runs for each request for each wavelength. So the overall complexity is $[C(n,2)-n] \times[O(n^2)\times W] \approx O(W\times n^4)$. For all practical purposes, we can assume W<<n. So, the overall complexity is $O(n^5)$.

Algorithm 5: Finding the shortest path on virtual topology has $O(n^2)$ complexity and it's repeated [C(n,2)-n] times for the worst case. So the overall complexity is $[C(n,2)-n]\times O(n^2) \approx O(n^4)$.

Thus the complexity of either POW or POT with multi-hop assignment is O(n^5).

VI. SIMULATION RESULTS

We evaluate the performances of our proposed algorithms by simulating random graphs of various sizes and densities. As we are considering both *Dense WDM* (DWDM) and *Sparse WDM* (SWDM) networks in this paper, we simulate the algorithms on the graphs having various densities. Our density gradations of random graphs are as follows: Graph of N number of nodes with density δ has $\lceil \delta\times(N-1) \rceil$ ($\delta \leq N/2$) number of edges placed randomly all over the graph. We also assume that a single lightpath can take care of all the traffic demands between two nodes of a network.

We consider the following performance metrics to compare the performances of the two RWA algorithms discussed above: Priority on Traffic (POW) and Priority on Wavelength (POT).

- *Total one-hop traffic* (h) is defined as the total amount of traffic allocated by one-hop lightpaths only in virtual topology.
- *Sigma Factor* (σ) denotes the average hop-length in the virtual topology. Numerically, it's the weighted mean of total traffic carried by a lightpath and the corresponding hop lengths on the virtual topology, i.e. if t_i and h_i denote the traffic and hop-length of lightpath of G_v on G_p, respectively,

$$\text{Sigma Factor } (\sigma) = (\textstyle\sum t_i \times h_i) / \sum t_i.$$

Note that $\sigma \geq 1$ and it is 1 when h_i is one (one-hop traffic) for all set of requests, i.e., we find a survivable virtual topology only through one-hop lightpaths. This also implies AT = T. $\sigma > 1$ implies there must have at least one multi-hops assignment in the virtual topology.

- *Blocking Probability* (b) is the amount of traffic not possible to be assigned with any wavelength by one-hop with respect to total traffic.
- *Wavelength Utilization* (u) is the % of total number of wavelengths used for one-hop assignment.
- *Number of multi-hops* (m) is defined to be the number of requests satisfied by multi-hops (wavelength conversion at nodes) in a virtual topology.

TABLE 1. PERFORMANCE METRICS FOR THE *SWDM* FOR PRIORITY ON TRAAFIC (*POT)* AT TOP AND PRIORITY ON WAVELENGTH (*POW)* AT BOTTOM.

No. of NODES (N)	**No. of WAVE-LENGTHS (|λ|)**	**DENSITY (δ)**	**SIGMA FACTOR (σ)**	**BLOCKED ONE-HOP TRAFFIC (%) (b)**	**WAVE-LENGTH UTILIZATION (%) (u)**	**NO. of MULTI-HOPS (m)**	**BUFFER SIZE (BS)**	**AVERAGE BS/NODE**
6	2	2	1.23	16.5	0.73	4	102	16.75
7	2	2	1.25	29.1	0.74	9	264	37.67
10	3	3	1.03	0.0	0.82	6	65	6.55
6	2	2	1.09	20.4	0.85	2	65	10.91
7	2	2	1.14	30.0	0.90	5	148	21.00
10	3	3	1.02	3.0	0.91	9	36	3.60

TABLE 2. PERFORMANCE METRICS FOR THE *DWDM* FOR PRIORITY ON TRAAFIC (*POT)* AT TOP AND PRIORITY ON WAVELENGTH (*POW)* AT BOTTOM.

No. of NODES (N)	**No. of WAVE-LENGTHS (|λ|)**	**DENSITY (δ)**	**SIGMA FACTOR (σ)**	**BLOCKED ONE-HOP TRAFFIC (%) (b)**	**WAVE-LENGTH UTILIZATION (%) (u)**	**NO. of MULTI-HOPS (m)**	**BUFFER SIZE (BS)**	**AVERAGE BS/NODE**
8	3	2	1.02	4.5	0.83	3	33	4.19
9	3	2	1.15	23.4	0.74	13	263	29.17
15	5	3	1.01	6.6	0.74	7	74	0.47
8	3	2	1.00	3.3	0.89	1	7	1.19
9	3	2	1.04	11.6	0.96	6	72	8.22
15	5	3	1.00	2.6	0.80	0	0	0.00

- *Buffer Size* (BS) is the total amount of traffic loads in all the nodes due to multi-hops assignments.

The tables above display the data obtained for the above performance parameters for the two RWA algorithms from the same simulation environment, for SWDM and DWDM. To divide the graphs into SWDM and DWDM, we have calculated the number of Effective WDM Links (EWL) which is defined as the links available for wavelength assignments in link-state matrix. Numerically, it's the product of number of wavelengths per link, number of links and density of a graph, i.e. EWL = ($W \times N \times \delta$). Higher the EWL corresponding to the number of nodes of the graph, higher is the density of the graph and it falls into DWDM.

It's clear from both the tables that, Wavelength utilization is always better for POW, as expected. This is because the algorithm utilizes the wavelengths one by one and after all the requests are exhausted for a wavelength, next wavelength is used. So with the number of wavelengths as constraint, POW is always a better choice to get one-hop traffic.

For SWDM, blocked one-hop traffic is always better for POT as expected. This is because of the fact that we assign heavy traffics for any wavelength for POT. On the other hand, Sigma factor, number of multi-hops and Buffer size suggest that POW has less number of multi-hops. Thus, without conversion capability, POW is again a better choice. For DWDM, the picture is even clearer. POW clearly stands for without wavelength conversion and POT stands for with wavelength conversion.

So we conclude that POW is a better candidate for WDM network *without wavelength conversion* capability in nodes and with a few numbers of wavelengths. This will give higher one-hop traffic. On the other hand, POT is a better choice for WDM network with wavelength conversion capability at nodes and some high priority traffics which need to be routed at any cost.

VII. CONCLUSIONS AND FUTURE WORK

In this paper we investigated the problem of Survivable Virtual Topology Routing [SVTR] under single fiber failure in WDM weighted mesh networks. We have proved necessary and sufficient conditions for the existence of survivable routes. Using that condition, we have proposed a polynomial algorithm to establish survivable routes using a single wavelength. After creating this survivable ring virtual topology, we have assigned additional lightpaths using the available wavelengths at the various links to increase the one hop traffic mainly, giving priority on traffic and wavelength, respectively.

Optical network is mainly used in WAN, which suffers from multiple fibers failure simultaneously in practical as it is spread over a big city, nation or more. So our future work consists of designing a survivable RWA algorithm, which provides continuous service even after multiple fiber links failure. Also, due to survivability, our algorithms provide less one-hop traffic than those of some well-known non-survivable RWA algorithms. Thus, there is again a place of improvement in this area.

ACKNOWLEDGEMENTS

The authors would like to thank the teaching and non-teaching staff members of the Computer Science and Engineering department of Haldia Institute of Technology where the ground work of this paper was done. They also thank Prof. S. Ghosh, Prof. I. Sengupta, IIT, Kharagpur and Dr. R. Datta, NERIST who introduced them to the problems of WDM Networks.

REFERENCES

[1] L. Sahasrabuddhe, S. Ramamurthy, B. Mukherjee, "Survivable WDM Mesh Networks", *IEEE*, Vol-21, No-4, April-2003.

[2] S. Yuan, Jason p. Jue, "Dynamic Lightpath Protection in WDM Mesh Networks under Wavelength Continuity Constraints", *IEEE Globecom*, 2004.

[3] D. Banerjee, B. Mukherjee, "A Practical approach for routing and wavelength assignment in largewavelength-routed optical networks", *IEEE Journal of Selected Areas in Communications*, vol. 14, no. 4, June 1996.

[4] B. Mukherjee, S. Ramamurthy, D. Banerjee, A. Mukherjee, "Some principles for designing a wide-area optical network", *IEEE/ACM Transactions on Networking*, vol. 16, no. 7, Sepetember 1998.

[5] A. Todimala, B. Ramamurthy, "Survivable Virtual Topology Routing under Shared Risk Link Groups in WDM Networks", *BROADNETS* '04.

[6] E. Modiano, A. Narula-Tam, "Survivable Lightpath Routing: A New Approach to the design of WDM-Based Networks", *IEEE Journal on Selected Areas in Communications*, vol. 20, no. 4, May 2002.

[7] E, Modiano, A. Narula_Tam, "Survivable routing of logical topologies in WDM networks", *IEEE INFOCOM* 2001.

[8] A. Sen, B. Hao, B. H. Shen and G. H. Lin, "Survivable routing in WDM networks logical ring in arbitrary physical topology", *Proceedings of the IEEE International Communication Conference ICC02*, New York, June 2002.

[9] H. Lee, H. Choi, S. Subramaniam, and H. A. Choi, "Survival Embedding of Logical Topology in WDM Ring Networks", *Proc. The 6th Joint Conference on Information Sciences (JCIS),* Duke University, March 2002.

[10] H. Lee, H. Choi, S. Subramaniam, and H. A. Choi, "Survival Logical Topology Design inWDMOptical Ring Networks", *Proc. 39th Allerton conference on communications, control, and computing,* UIUC, Oct. 2001.

[11] http://www.taygeta.com/random/boxmuller

[12] http://www.geocities.com/dharwadker/hamilton/

[13] A. Mokhtar, M. Azizoglu, "Adaptive Wavelength Routing in All-Optical Networks", *IEEE/ACM Transactions on Networking*, Vol-6, No.2 April 1998.

APPENDIX - I

If $G_v = (V, E_v)$ is the virtual topology then a cut is a partition of the set of nodes V into two parts T and V-T. We define a *cut-set* as a set of edges in E_v with one endpoint in T and the other in V-T and denote it as CUT(T,V-T). So, from *Menger's theorem*, a virtual topology G_v is 2-connected if and only if every non-trivial cut set CUT(T,V-T) contains at least two edges.

The following Lemma gives a necessary and sufficient condition for the survivable routing on an arbitrary virtual topology.

Lemma 1

Let E(s,t) be the set of physical links used by lightpath (s,t) of the virtual topology G_v. A routing is survivable if and only if for every cut set CUT(T,V-T) of the virtual topology G_v, $\cap E(s,t)=\phi$ where $(s,t) \in$ CUT(T,V-T). This condition requires that no single physical link is shared by all virtual links belonging to a cut set of the virtual topology. That means all the lightpaths belonging to a cut set can not be routed on the same physical link. This condition must hold for all cut sets of the virtual topology.

Proof: To prove the necessary condition, assume that the routing is survivable and still $\cap E(s,t) \neq \phi$. i.e. there exist a physical link $e \in \cap E(s,t)$ where $(s,t) \in$ CUT(T,V-T) and $e \in E_p$ where $G(V,E_p)$ is the physical topology. So the failure of the fiber optical link e in physical topology will disconnect all the lightpaths belonging to the cut set CUT(T,V-T) in virtual topology and split the virtual topology G_v into more than one component. So to make the virtual topology survivable, $\cap E(s,t)= \phi$ where $(s,t) \in$ CUT(T,V-T).

To prove the sufficient condition, assume that $\cap E(s,t)= \phi$ where $(s,t) \in$ CUT(T,V-T). Also assume that a physical link $e \in E(p,q)$ where $(p,q) \in E_v$. Let $S \subseteq E_v$ and $S= \cup (p,q)$. So S is the set of lightpaths (edges of the virtual topology G_v) which are routed through the physical link e. So the failure of the link e will remove the set of edges S from the virtual topology G_v. Therefore the new virtual topology will be $G_{v1}(V, E_v\text{-}S)$. Now if $\cap E(s,t)= \phi$ is satisfied then it is obvious that S is not a cut set, since G_{v1} is a connected graph. So the survivability of the virtual topology G_v is preserved.

The following corollary to Lemma 1 gives a necessary and sufficient condition for the survivability in a bi-directional ring virtual topology.

Corollary 1

A virtual ring topology G_v is survivable if and only if no two lightpaths in G_v share the same physical link in G_P.

Proof: To prove the necessary condition, assume that the ring is survivable, yet virtual links e_1 and e_2 in G_v share a physical link t in G_P. So the failure of the link t will split the ring into more than one component. The ring does not remain survivable anymore.

Now to prove the sufficient condition, assume that any two virtual links of the virtual topology is not sharing any physical link. Let (e_1, e_2) is any cut set of G_v. From the property of the ring topology, we know that any two links $(e_1, e_2) \in E_v$ of a virtual ring topology G_v constructs a cut set. Therefore failure of a physical link t will either affect e_1 or e_2 or none them. So the failure of any physical link leaves the virtual topology connected.