Downloaded from orbit.dtu.dk on: Aug 22, 2019

DTU Library

An algorithm for link restoration of wavelength routing optical networks

Limal, Emmanuel; Stubkjær, Kristian

Published in:
Conference Paper on IEEE International Conference on Communications, 1999

Link to article, DOI:
10.1109/ICC.1999.765620

Publication date:
1999

Document Version
Publisher's PDF, also known as Version of record

Link back to DTU Orbit

Citation (APA):
Limal, E., & Stubkjær, K. (1999). An algorithm for link restoration of wavelength routing optical networks. In Conference Paper on IEEE International Conference on Communications, 1999 (Vol. 3, pp. 2055-2061) https://doi.org/10.1109/ICC.1999.765620

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

An algorithm for link restoration of wavelength routing optical networks

Emmanuel Limal and Kristian E. Stubkjær

Research center COM, Technical University of Denmark, Building 348, DK-2800 Lyngby, Denmark
Tel.: +45 45 88 14 44, Fax.: +45 45 93 65 81, E-mail: el@com.dtu.dk

***Abstract*—We present an algorithm for restoration of single link failure in wavelength routing multihop optical networks. The algorithm is based on an innovative study of networks using graph theory. It has the following original features: it *(i)* assigns working and spare channels simultaneously, *(ii)* prevents the search for unacceptable routing paths by pointing out channels required for restoration, *(iii)* offers a high utilization of the capacity resources and *(iv)* allows a trivial search for the restoration paths. The algorithm is for link restoration of networks without wavelength translation. Its low complexity is studied in detail and compared to the complexity of a classical path assignment algorithm. Finally, we explain how to use the algorithm to control the restoration path lengths.**

***Index terms*—network, link restoration, graph theory and Depth First Search algorithm.**

I. Introduction

The large bandwidth of optical fibers makes the transmission of terabit per second possible by multiplexing several wavelengths on one fiber. Wavelength routing multihop networks are therefore expected to carry tens of Terabit per second on a single link, where a link is a physical connection between two consecutive nodes that consists of a group of fibers, each containing a number of wavelength channels. To avoid that a huge amount of data is affected by a single link failure, restoration strategies are required. Presently, restoration is provided in the SDH layer but the introduction of optical cross-connects will make fast restoration in the optical domain possible and desirable. We consider consequently the problem of assigning just enough *spare* capacity (in terms of channels) in a network to ensure restoration of a single link failure. This problem has gained considerable attention recently [1-5] and algorithms using the classical integer linear programming (ILP), linear programming (LP) [1][4-5] and simulated annealing (SA) [4] methods are among the state of the art procedures to solve this NP hard problem (up to today, no NP hard problem has been solved with a polynomial-time algorithm) [2].

In a localized multihop optical network with many nodes, the optical cross-connect architectures will probably be simple to ensure low cost and therefore contain small space switches and no wavelength converters. In such networks, the ILP, LP and SA methods are inappropriate because of their high complexity and the integer values of capacities (i.e. the number of channels is an integer). We propose the *BONRA* (*Bi-directional Optical Network Restoration Algorithm*) to allocate spare capacity in networks. The BONRA can be considered as a class of algorithms but in this paper, we restrain this name to the case of *link restoration* (the signals are re-routed between the end nodes of the failed link as opposed to *path restoration*) of single link failure, for networks without wavelength translation. The BONRA has a low complexity and is, consequently, most practical for dynamic management of networks with evolving traffic demand.

The paper is organized as follows. In section II, the similarity between networks and graphs is recalled and a new terminology proper to restoration is introduced. We present in section III, the graph theory study of single link failure restoration for networks without wavelength translation. Then, the BONRA is described, investigated and discussed in section IV, V and VI, respectively.

II. Graph Theory

In this section, we recall terms from graph theory, point out their equivalence from a network perspective and introduce a new terminology intended for restoration.

A. *Graph terminology* [6]

A graph $G=(V,E)$ consists of a set of $|V|$ *vertices* (singular, *vertex*) and a set of $|E|$ *edges*. Each edge $e_{ij}=(i,j)$ is associated to a pair of vertices the same way that a link in a network is connecting two nodes. In the graph representation of G, vertices are points and edges are lines bounding them. In this work, we consider *undirected* graphs where an edge e_{ij} can be indifferently defined as the vertex pair (i,j) or (j,i). For the network it means that a connection is made from a node i to a node j and vice-versa. The *degree*, d_i, of vertex i is the number of edges *incident on i*. An edge is said to be incident on its *end-vertices*.

A *path* is an alternating sequence of vertices and edges $<1,e_{12},2,\dots,k\text{-}1,e_{k\text{-}1,k},k>$ both starting and ending with a vertex, such that no vertex and no edge appear twice. A *circuit* is a *closed* path, its end vertices are identical (e.g., $k=1$). For simplicity we define all the vertices of a path p that are not its end-vertices, its *internal vertices*. We call a *line* $L=(V_L,E_L)$ of a

0-7803-5284-X/99/$10.00 © 1999 IEEE.

Authorized licensed use limited to: Danmarks Tekniske Informationscenter. Downloaded on May 31,2010 at 07:11:28 UTC from IEEE Xplore. Restrictions apply.

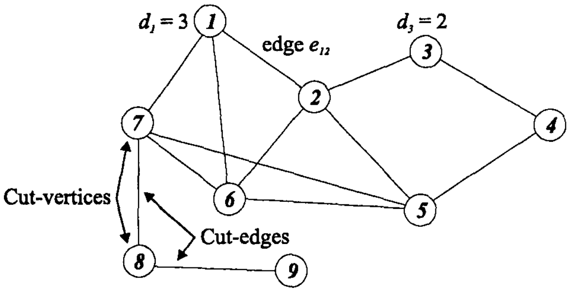


Fig. 1: Graph with nine vertices, thirteen edges, two lines: <2,e_{23},3,e_{34},4,e_{45},5> and <7,e_{78},8,e_{89},9>, two cut-vertices: vertices *7* and *8*, two cut-edges: e_{78} and e_{89} and a bi-connected component ({*1,2,3,4,5,6,7*}, {e_{12},e_{16},e_{17},e_{23},e_{25}, e_{26},e_{34},e_{45},e_{56},e_{57},e_{67}}).

graph G, a path whose internal vertices have all exactly a degree 2 in G and whose end-vertices both have a degree different than 2, as illustrated in fig. 1.

A *tree T* of G is a sub-graph of G that do not contain any circuit. A *spanning tree* of G is a tree having all vertices of G. A *cut-vertex* and a *cut-edge* of a *connected* graph G are respectively a vertex and an edge whose removal disconnect G. A *bi-connected component* of G is a maximal connected sub-graph of G with no cut-vertices (fig. 1).

B. An optical network as a collection of graphs

Wavelength routing optical networks have different number of wavelengths multiplexed per fiber. If no wavelength translation is provided at the network nodes, there is a corresponding topology in the network for each wavelength. In fig. 2, a network with two wavelengths is shown and the topologies corresponding to each wavelength are illustrated beneath it.

In such a network, restoration of a disrupted signal λ_1 can only occur in the wavelength λ_1 topology (fig. 2b). Thus, there are as many restoration sub-problems as there are independent wavelengths. Unless specified otherwise, we consider now the graph equivalent to a specific wavelength topology, say λ. Each edge (i,j) of the graph corresponds to the channel of the link (i,j) at the wavelength λ. Therefore, we use indifferently *edge* or *channel* depending on the context.

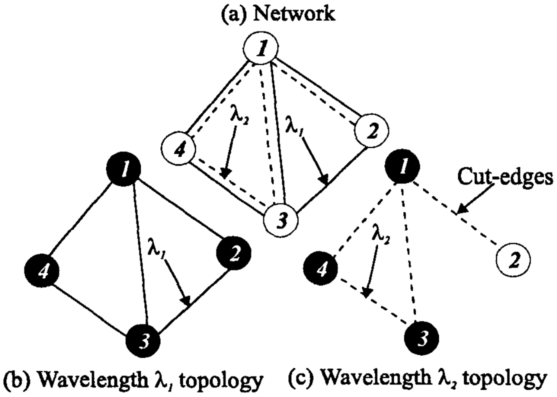


Fig. 2: In (a), a network with two different wavelengths, 4 nodes and 5 links is shown. There are two specific topologies, (b) and (c), for λ_1 and λ_2, respectively. Considering each wavelength topology as a graph, the black vertices are elements of a bi-connected component.

C. The problem

Assuming that a network is given and its collection of wavelength topologies is identified, we consider the problem of managing the network capacity resources assuming the traffic demand is dynamically evolving. In such conditions, we need to develop the theory to both assign capacity for each new traffic connection requested and reserve capacity for restoration purposes. Moreover, the released available capacity must be identified whenever an old traffic connection is canceled. In the next section, we clarify these aspects by introducing a new terminology that classifies channels (edges) in four *states*.

D. Restoration terminology

One new feature of the BONRA is the additional information it provides about the network channels. This information is comprised in the following states:

Working a channel is *working* if it is used to transmit traffic (that must be restored in case of link failure).

Spare a channel is *spare* if it is *not* working.

Forbidden a channel is *forbidden* if it is spare and cannot be set to a working state without preventing other working channels to be restored.

Available a channel is *available* if it is spare and not forbidden (i.e. if it is set to a working state, all working channels will still be restorable).

The particular feature of the *forbidden* and *available* states is the information they provide on the consequences of setting a spare channel in a working state. This is illustrated in fig. 3(a) where a network topology $G_\lambda=(V_\lambda,E_\lambda)$, at a specific wavelength λ, is given and the *working*, *forbidden* and *available* edges are identified. In fig. 3(b) and (c) G_λ is shown after an available and a forbidden edge is set to a working state, respectively. We define the *spare graph*, $G_{\lambda,S}$, of a graph G_λ as the sub-graph of G_λ that contains all its spare edges and their end-vertices. The *available graph*, $G_{\lambda,A}$, of a graph G_λ is similarly the sub-graph of G_λ that contains all its available edges and their end-vertices. For simplicity, we suppress the index λ. Consequently from these definitions and the example of fig. 3, we can conclude that only available channels can be use to assign working paths in a network. Moreover, available and forbidden edges must be identified each time a spare channel is set to a working state or vice versa.

Authorized licensed use limited to: Danmarks Tekniske Informationscenter. Downloaded on May 31,2010 at 07:11:28 UTC from IEEE Xplore. Restrictions apply.

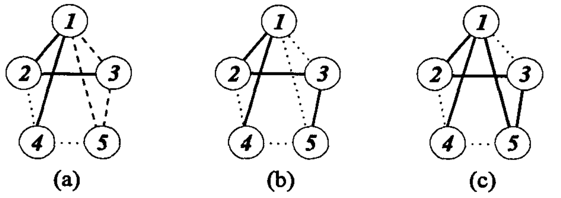


Fig. 3. In (a), a graph G is shown. Working edges, forbidden and available edges are full, dotted and dashed line, respectively. The three edges $\{e_{13},e_{35},e_{51}\}$ are available in (a). For example, when the *edge* e_{35} is set to a working state, all the working edges are still restorable (b). By definition, *edge* e_{35} is available in (a). However, when the *edge* e_{15} of (b) is set to a working state, there are *working edges* (e.g., e_{14}) that are not restorable (c). By definition, *edge* e_{15} is thus *forbidden* in (b).

III. Link Restoration Theory

In the following sections we briefly present the theory required to ensure that all working channels are restorable and to recognize all forbidden and available channels. Consider the topology of a network, at a specific wavelength, given by $G=(V,E)$.

A. Topology requirements

In the graph $G=(V,E)$, link restoration of a working edge e is only possible if e belongs to a bi-connected component. The proof is obvious. In fig. 2c, e_{12} is a cut-edge (it do not belong to any bi-connected component) and can, therefore, not be restored. Consequently, for a given graph G, a primary requirement for link restoration is the identification of all bi-connected components of G. But this is not enough.

Link restoration is not always directly applicable in *lines*. Indeed, link restoration is not possible in a line if there are *more* than one working edge in the line. The reason is that to restore a working edge e of a line L, there must be a *spare path* (path of spare edges) between the end-vertices of e. And, by definition of a line, such a path requires that all edges of L-$\{e\}$ are spare. Thus no other edge than e must be working in the line. To overcome this problem, it is sometimes possible to simply use path restoration in lines. However, to avoid a too extensive discussion, we assume in the following that G is a bi-connected graph with no lines.

B. Link restoration of working channels

The requirements for link restoration of all working edges of a graph G are simply that:

(i) the *spare graph*, $G_S=(V_S,E_S)$, of the graph G must be connected

(ii) the *spare graph*, $G_S=(V_S,E_S)$, must contain all vertices of G, (i.e. $V_S=V$).

This can be proved by graph theory by showing that all vertices must belong to the spare graph and that there must be a spare paths between each pair of end-vertices of working edges. As an example, a close look at fig. 3(c) shows that working edges are not restorable because G_S is not connected. The conditions (i) and (ii) to ensure single link failure restoration are sufficient to construct a restoration algorithm. However, in the next section we give the theory behind the concept of *forbidden* and *available* edges in order to shorten the restoration algorithm running time.

C. The properties of the spare graph

By definition, forbidden edges are spare edges that cannot be used to route traffic if link restoration must be guarantied. It would, therefore, be convenient to identify them in an easy manner. Actually, the criterion to discern them is simple and can be implemented simultaneously to the criteria (i) and (ii) for restoration, adding no complexity to the algorithm. From graph theory it can be proved that:

(iii) A *forbidden* edge of a graph G is a cut-edge of the spare graph, $G_S=(V_S,E_S)$, of G

(iv) An *available* edge of a graph G is an edge belonging to a bi-connected component of the spare graph, $G_S=(V_S,E_S)$, of G

At this point, we have the whole theory needed to develop the BONRA.

IV. Algorithm

The BONRA makes use of the Depth First Search (DFS) algorithm [7] that can be adapted to identify all bi-connected components and cut-edges, thus, available and forbidden edges of G_S [6-7]. The complexity of an algorithm is the rate of growth of its running time. As an example, The DFS has a complexity $O(E_S)$. It means that the running time of the DFS for a graph $G_S=(V_S,E_S)$ is mainly driven by the number of edges $|E_S|$ of G_S.

A. The DFS algorithm

The DFS algorithm for undirected graphs makes use of a systematic method, starting with any (source) vertex, for exploring a graph that is assumed without self-loop [7]. Applied to a spare graph G_S of a graph $G(V,E)$, the DFS investigates all vertices and edges in a tree manner. If the DFS sees the $|V|$ vertices of G_S within *one* tree, then G is connected and contains all vertices of G as required by link restoration requirements (i) and (ii).

B. The Shortest path

We use Yen's algorithm [8] to calculate the k^{th} shortest path, $p_{ij,k}$, to establish connection between i and j. A path is shorter than another if it contains less edges. We call k_{max} the limit for k (i.e. $k \leq k_{max}$).

Authorized licensed use limited to: Danmarks Tekniske Informationscenter. Downloaded on May 31,2010 at 07:11:28 UTC from IEEE Xplore. Restrictions apply.

C. The BONRA

To reduce the text we use the notation $G_{\lambda,i,A}$ where λ is the wavelength index, i is the bi-connected component index in G_λ and A (or S) refer to the *available* (or *spare*) sub-graph of $G_{\lambda,i}$. If no index i is used, both bi-connected components and cut-edges are considered. If no A (or S) index is used, all edges are considered indifferently of their states.

The BONRA procedure contains 7 steps as follows:

1. Identify using the DFS algorithm, for each wavelength topology G_λ, all the bi-connected components $G_{\lambda,i}$ of G_λ. Set all edges of each G_λ to an *available spare* state. Choose k_{max}.
2. Select a connection (i,j) to be assigned. If a connection has been canceled, set the corresponding working edges to a spare state and go to step 7.
3. Select a wavelength λ that has not been tried yet to assign (i,j). If all the wavelength have been tried the consider connection cannot be assign under the single link failures restoration guaranty.
4. Compute the k_{max} shortest paths between vertex i and j in the available graph $G_{\lambda,A}$ of G_λ. If no such path exist, go to 3.
5. Try the shortest path not yet tried with wavelength λ, to assign (i,j).
6. For each bi-connected component $G_{\lambda,i}$ of G_λ verify, using DFS, that its spare graph $G_{\lambda,i,A}$
   (i) is connected
   (ii) contains all vertices of $G_{\lambda,i}$

   If (i) and (ii) are *true* go to 7. Else if (i) and/or (ii) are false and all the shortest path have not been tried, got to 5. Finally, if (i) and/or (ii) are false and all the shortest paths have been tried at the wavelength λ, go to 3.
7. For each bi-connected component $G_{\lambda,i}$ of G_λ set the cut-edges of its spare graph, $G_{\lambda,i,Spare}$, to a forbidden state and the other spare edges to an available state. Go to 2.

In step 3, there are three classical methods to select a wavelength;

(i) *Random* (select a random wavelength)
(ii) *Most used* (select the wavelength of the topology with least spare edges)
(iii) *First fit* (select the wavelength of lower index),

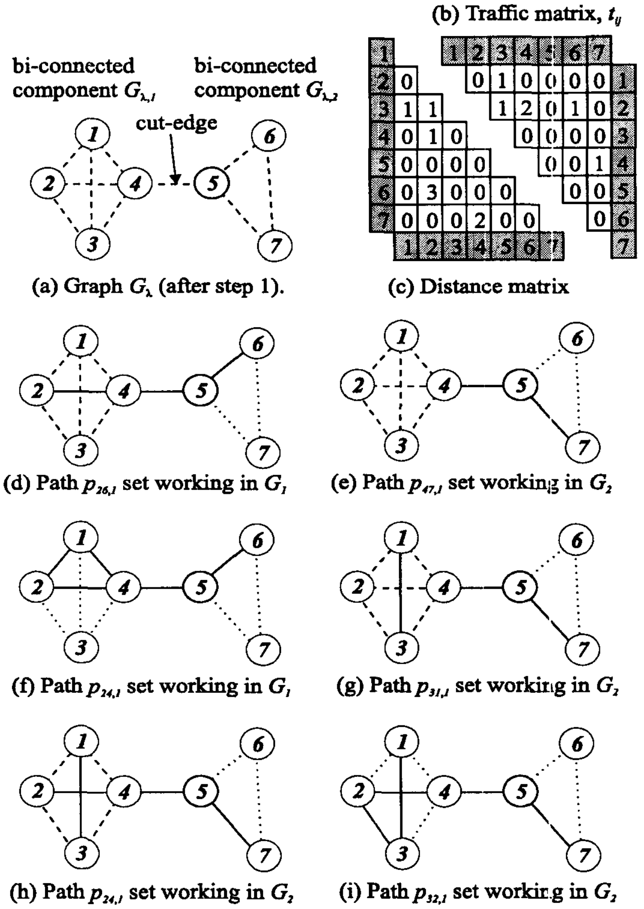
Their performances are compared in section V.*C*. To observe the influence of the order in which connections have to be assigned we consider the two following connection arrival order;

(i) *Random order*
(ii) *Selective order:* Select the connection (i,j) that has the longest shortest paths in G.

Note that a random order is best to reflect a dynamic traffic pattern.

D. Illustration of the BONRA

The functions of the BONRA are illustrated for the topology of fig. 4a and the traffic matrix of fig. 4b. The example is for selective order of connection and the *first fit* wavelength selection method. Fig. 4d: connection $(2,6)$ is chosen first as it has the longest shortest path with three edges. The first (k=1) shortest path $p_{26,1}$ is assigned with the first wavelength (in $G_{1,A}$).


bi-connected component $G_{\lambda,1}$
bi-connected component $G_{\lambda,2}$
cut-edge
1
2
3
4
5
6
7

(b) Traffic matrix, t_{ij}

1	2	3	4	5	6	7	
	0	1	0	0	0	0	1
		1	2	0	1	0	2
			0	0	0	0	3
				0	0	1	4
					0	0	5
						0	6
							7

1							
2	0						
3	1	1					
4	0	1	0				
5	0	0	0	0			
6	0	3	0	0	0		
7	0	0	0	2	0	0	
	1	2	3	4	5	6	7

(a) Graph G_λ (after step 1).

(c) Distance matrix

(d) Path $p_{26,1}$ set working in G_1

(e) Path $p_{47,1}$ set working in G_2

(f) Path $p_{24,1}$ set working in G_1

(g) Path $p_{31,1}$ set working in G_2

(h) Path $p_{24,1}$ set working in G_2

(i) Path $p_{32,1}$ set working in G_2

Fig. 4. Illustration of the functions of the BONRA for the topology (a) containing 7 vertices and the traffic matrix (b). The matrix (c) gives the shortest path distances, in edges, used for the selective order of connection. Dotted, dashed and plain edges are forbidden, available and working edges, respectively. Read $p_{47,1}$ as the first shortest path between vertices 4 and 7. Read G_1 and G_2 as the graph at wavelength λ_1 and λ_2. (d)-(i) gives the order and the wavelengths in which the path are assigned. As an example, (f) shows the status (after step 7) of the graph of wavelength 2, when paths $(2,6)$, $(4,7)$ and $(4,2)$ have already been assigned.

Authorized licensed use limited to: Danmarks Tekniske Informationscenter. Downloaded on May 31,2010 at 07:11:28 UTC from IEEE Xplore. Restrictions apply.

The two spare graphs $G_{1,1,S}$ and $G_{1,2,S}$ are still connected and contain all vertices of the bi-connected components $G_{1,1}$ and $G_{1,2}$, *respectively.* The path is accepted and forbidden edges are identified.

Fig. 4e: connection (*4,7*) is chosen as its shortest path has two edges. None of the k_{max} path (*4,7*) can be set in $G_{1,A}$. Another wavelength is required. The first shortest path $p_{47,1}$ is assigned with the second wavelength (in $G_{2,A}$). The spare graphs $G_{2,1,S}$ and $G_{2,2,S}$ are still connected and contain all vertices of $G_{2,1}$ and $G_{2,2}$, *respectively.* The path $p_{47,1}$ is accepted and forbidden edges are identified.

Fig. 4f-i show the graph status subsequent to step 7, after the assignment of path (*4,2*), (*3,1*), (*2,4*) and (*3,2*), respectively.

E. Complexity of the algorithms

The BONRA is assigning both working path and spare resources simultaneously as opposed to the classical ILP, LP and SA methods that allocate spare resources once all working paths are assigned. For this reason, we discuss the complexity of the BONRA per working path. Schematically, the BONRA chooses a wavelength, finds a shortest path in the corresponding topology, verifies using the DFS algorithm that restoration is possible and if not, repeat these steps until a satisfactory solution is found. For each wavelength λ tried, the shortest path is computed within the *available graph* $G_{\lambda A}(V_{\lambda A},E_{\lambda A})$ and has thus a complexity of $O(V_{\lambda A}{}^2)$ [9]. The DFS algorithm is applied to the *spare graph* $G_{\lambda S}(V_{\lambda S},E_{\lambda S})$ obtained from step 5 and has thus a complexity of $O(E_{\lambda S})$ [6]. The BONRA's complexity is, therefore, $O(\Sigma_\lambda[V_{\lambda A}{}^2+E_{\lambda S}])$ per assigned path. By comparison, an algorithm that assigns working paths without restoration requirements has a complexity of $O(\Sigma_\lambda[V_{\lambda,S}{}^2])$ per path. The two algorithm complexities are comparable, as illustrated in table 1.

Table 1. Illustration of the low complexity of the BONRA (most used selection of λ and selective order of connection).

Average over all λ's	BONRA-LR	No Restoration
Shortest path	$V_A = 14.5$	$V_S = 16.34$
DFS algorithm	$E_S = 27.8$	No
wavelengths tried	$\Sigma_\lambda = 9.07$	$\Sigma_\lambda = 7.01$
Relative complexity	110.3 %	100 %

V. Results

In this section, the BONRA is investigated with regards to the capacity requirement, k_{max}, wavelength selection methods and connection orders. We apply the BONRA to dimension the number of wavelengths required to assign an uneven traffic matrix demand as in [10] for a network topology of 19 nodes with degree between 2 and 7 and 39 links [11].

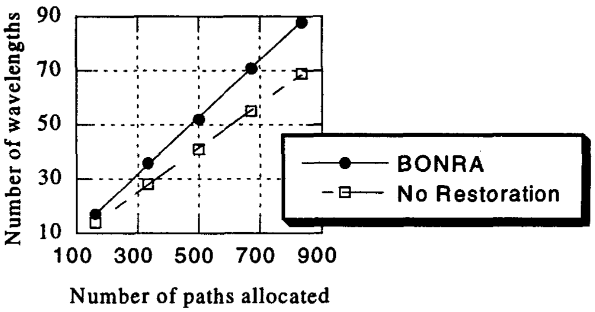


Fig. 6. Number of wavelengths required by the BONRA and a simple working path assignment algorithm versus the number of paths allocated. Lines are linear fits.

A. Requirement for spare capacity

For a bi-connected wavelength topology $G=(V,E)$, it can be proven that the minimum spare graph (containing least edges) $G_S=(V_S,E_S)$ satisfying conditions (i) and (ii) for link restoration is a *spanning tree* of *G*. It is known from graph theory that such a spanning tree has $|V|-1$ edges. Thus, under the requirement of single link failure restoration, the maximum number of working edges in $G(V,E)$ is $|E|-|V|+1$. For the network studied, the maximum fraction of working edges is $1-|V|/|E|+1/|E|=53.8$ %. However, the number of wavelengths required for transmission and restoration using the BONRA is only about 25 % higher than the number of wavelengths required for transmission. The reason is that the BONRA tend to fill, with spare capacity, links with poor working capacity. In fig. 5, the required number of wavelengths is computed for a scaling of the traffic matrix from 1 to 5 and for both the BONRA and the working path assignment algorithm. The wavelengths and paths are selected randomly.

B. The shortest path is the best

Because the BONRA identifies forbidden channels it should be most efficient with first shortest paths (k_{max}=1). The BONRA has been used to dimension the network for k_{max}=1 to 5, assuming random, most used and first fit wavelength selection

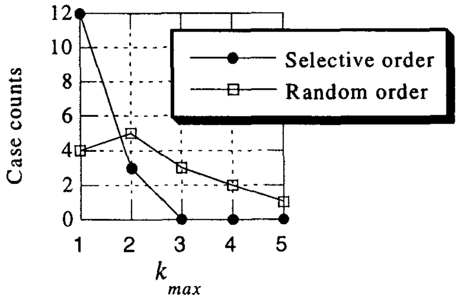


Fig. 5. Distribution of the lowest number of wavelengths required versus the maximum number of shortest paths k_{max}.

Authorized licensed use limited to: Danmarks Tekniske Informationscenter. Downloaded on May 31,2010 at 07:11:28 UTC from IEEE Xplore. Restrictions apply.

method. For each of these 15 cases, both random and selective order of connections were examined.

Fig. 6 gives the distribution of the lowest number of wavelengths required versus k_{max}. For the selective order of connections, the lowest number of wavelength is obtained in 12 cases, out of the 15, with k_{max}=1 and in the 3 other cases with k_{max}=2. For random order of connections, the tendency is also that the number of wavelengths required decrease as k_{max} is smaller.

C. How to choose the wavelength

We have presented three traditional methods called *random*, *most used* and *first fit*- to select the wavelengths used to route paths. The deviation, in terms of wavelength requirement, between these methods is negligible. However, the use of random selection of wavelength can reduce the computing time by a factor of two.

D. The BONRA applied to network management

When managing a network restoration in a dynamic environment, the connections have to be assigned in the order they are requested. This can be simulated by the random order of connection request. In fig. 7, we compare it to the selective order of connection assuming that the most used wavelengths are chosen and k_{max}=1. The BONRA use the wavelengths less efficiently for random order of connection. The deviation in the number of wavelengths is within 7%. Thus, if the traffic demand is known in advance, it is advised to have a good selective order of connections.

VI. Discussion

In some networks there can be requirements for the length of restoration paths. In the following sections, we explain how to adapt the BONRA to control restoration path length without changing the algorithm complexity.

A. The BONRA adaptability to network size

The greatest advantage of the BONRA is its low complexity which makes it attractive to dynamic management where it should be used each time new traffic demands arise or old traffic demands are canceled. When an operator wants to transmit traffic with guaranty of restoration, he does not need to know the restoration path at the time he assigns the connection. The BONRA is fast because it gives the restoration guaranty without calculating the restoration path. However, a simple Floyd-Warshall shortest path algorithm [12] can be used afterwards to determine the restoration paths.

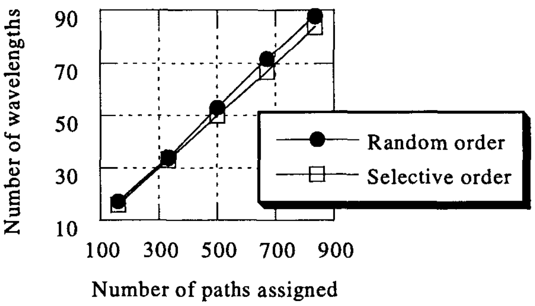


Fig. 7. Behavior of the BONRA for different orders in which the connections are requested.

The BONRA is ideal for networks where restoration paths with many hops are physically possible, that is in local area network. Indeed, the benefit of its low complexity increase as the number of network nodes and links increase. For networks where link distances are long and restoration path with a few hops required, it is possible to sub-divide the network topology in arbitrary bi-connected sub-networks and apply the BONRA to each of them. The restoration path lengths are then bounded by the sub-network sizes such that, in the worst case, the longest restoration path will have less hops than the sub-network has nodes. Moreover, the management is simplified as signaling is constrained in each sub-network. Whether the BONRA is applied to the whole network topology or to all of its sub-networks, its complexity is similar.

B. Comparison to other algorithms

As mentioned earlier, the main advantage of the BONRA is its low complexity and therefore suitability for dynamic management. Its simplicity is at the cost of a lower flexibility. But, the BONRA can be adapted to path restoration, path disjoint restoration, node restoration and even to restoration management of network with space or wavelength translation. In all cases the integer values of capacity is intrinsic to the BONRA. Further work is currently carried out to compare the BONRA for network with wavelength translation and ILP and SA algorithms.

VII. Conclusion

We have presented the BONRA, an algorithm for link restoration of single link failures in bi-directional networks. The algorithm is based on a graph theory study of networks and offers a precise knowledge of the channel states in the network. We have developed the required theory and illustrated the steps in the BONRA. It is ideal for dynamic management of restoration as it simultaneously assigns traffic paths and spare resources. The BONRA is practical for multihop wavelength routing networks of moderate size where restoration paths with many-hops are feasible. Finally, we discussed briefly how to use the BONRA in large area network and its advantages compared to other classical algorithms.

VIII. References

[1] H. Sakauchi, Y. Okanoue and S. Hasegawa, "*Spare-channel Design Schemes for self-healing Networks*", IEICE Trans. Commun., Vol. E75-B, No. 7, pp. 624-633, July 1992.

[2] B.. D. Venables, W. .D. Grover, and M.H MacGregor, "*Two strategies for space capacity placement in mesh restorable networks*", in

Authorized licensed use limited to: Danmarks Tekniske Informationscenter. Downloaded on May 31,2010 at 07:11:28 UTC from IEEE Xplore. Restrictions apply.

proceedings of ICC'93 conference, Vol. 1, pp. 267-271 Geneva, Switzerland, May 1993.

[3] E. Limal and K.E. Stubkjær: "*Upper bound for working capacity utilization in resilient optical networks using link restoration*", submitted to *Communication letter* in June 1998.

[4] B. Van Cænegem, N. Wauters and P. Demeester: "*Spare capacity assignment for different restoration strategies in mesh survivable networks*", in proceedings of ICC'97 conference, Vol. 1, pp. 288-292, Montréal, Canada, June 1997.

[5] M. Herzberg and S. J. Bye, "*An optimal Spare-Capacity Assignment Model for Survivable Networks with hop limits*", in proceedings of ICC'97 conference, Vol. 1, pp. 406-410, Montréal, Canada, June 1997.

[6] K. Thulasiraman and M. N. S. Swamy: "*Graphs: Theory and algorithms*", Wiley-Interscience, USA, 1992.

[7] R. E. Tarjan: "*Depth-First Search and Linear Graph Algorithms*", SIAM J. Comput., Vol. 16, pp. 146-160, 1972.

[8] J. Y. Yen: "*Finding the k shortest loopless paths in a network*", in Management Science, Vol. 17, No. 11, july, 1971.

[9] T. H. Cormen, C. E. Leiserson and R. L. Rivest: "*Introduction to algorithms*", Massachusetts Institute of Technology, USA, 1990.

[10] A. Chbat *et al*: "Toward wide-scale all-optical transparent networking: the ACTS "Optical Pan-European Network (OPEN)" project", invited paper to be publish in *Journal of selected areas on communication*, 1998.

[11] E. .Limal, B. Mikkelsen and K.E. Stubkjær: "*Building WDM wide area resilient transport network from 4-node semi-mesh and mesh sub-networks*", in proceedings of ICC'97, Vol. 3, pp. 1311-1318, Montréal, Canada, June 1997.

[12] R. W. Floyd, "Algorithm 97: Shortest paths," Comm. ACM, Vol. 5, p. 345, 1962.

Authorized licensed use limited to: Danmarks Tekniske Informationscenter. Downloaded on May 31,2010 at 07:11:28 UTC from IEEE Xplore. Restrictions apply.