Survivability Aware Routing of Logical Topologies: On Thiran-Kurant Approach, Enhancements and Evaluation

Muhammad S. Javed, Krishnaiyan Thulasiraman, Matthew A. Gaines
School of Computer Science,
University of Oklahoma, Norman, OK, USA
{javed, thulasi}@ou.edu

Guoliang Xue
Department of Computer Science and Engineering,
Arizona State University, Tempe, AZ, USA
xue@asu.edu

***Abstract*--Wavelength Division Multiplexing (WDM) can increase the carrying capacity of an optical network without laying additional fibers. However, a disruption in such a high speed and high capacity network can quickly impact the entire network. A fast protection and restoration recovery mechanism is needed to provide uninterrupted data delivery. Implementing IP directly over a WDM optical network, using optical crossconnects and IP routers, is emerging as the preferred method to efficiently utilize the enormous bandwidth offered by WDM networks. However, in such networks, a single link failure in the WDM layer can affect multiple links in the IP layer, which may greatly degrade data delivery. Several solutions have been proposed in the literature to avert such a scenario. These solutions mostly focus on finding paths for the IP connections in the WDM layer in such a way that the failure of a single WDM link does not disconnect the IP topology. Such a mapping is called "survivable". Due to the NP-completeness of the problem, various heuristics based on ILP formulations, tabu search, and shortest path variants have been proposed in the literature. In this paper we study a recent approach by Thiran and Kurant, point out certain attractive features and difficulties with this approach, and present enhancements to their basic approach to achieve better fault coverage and to add robustness to the survivable routing schemes. We provide simulation results evaluating the new heuristics.**

***Index Terms*—Logical topology, survivable routing, fault tolerance, protection interoperability, optical networks.**

I. INTRODUCTION

Implementing IP directly over the WDM optical networks has emerged as the current trend to effectively utilize the enormous bandwidth offered by optical networks [1]. This arrangement makes it possible to run a host of user applications on top of the IP layer [1]. To provide uninterrupted service, such a network should be able to tolerate failures. A network capable of providing such service is referred to as a survivable network. There are two widely accepted methodologies for providing this capability, namely protection and restoration. Protection requires pre-computed alternate paths for the failed components, whereas restoration requires computation of alternate paths at the time of failure [2].

The choice between protection and restoration mechanisms depends on several factors such as speed of recovery, the network layer under consideration, and the cost involved in terms of spare capacity and equipment allocation. Restoration mechanisms exhibit slower recovery speed but are considered capacity efficient, since resources are allocated only at the time of failure [1]. Protection mechanisms require dedicated capacity, but offer extremely fast recovery times usually less than 100 milliseconds [1]. Providing survivability at the WDM layer requires installation of additional fibers to provide diverse routes, which is difficult and expensive due to geographical and right of way limitations [1]. These limitations have resulted in significant interest in IP-over-WDM model, which utilizes IP routers and optical crossconnects to provide uninterrupted data delivery, as an alternative to providing survivability at the physical layer.

Implementing IP-over-WDM requires mapping an IP topology (usually called *logical topology*) over the physical WDM topology [2]. The mapping of each IP (logical) link involves finding a *lightpath* in the WDM network connecting the two end points of the logical link [3]. Usually a single fiber carries many lightpaths and all of them get disconnected when the physical link carrying them fails. However, if after the failure of a physical link/links, the IP topology remains connected, then the IP routers can find alternate routes for affected lightpaths after detecting the failure [2]. Such a mapping of IP topology on the physical WDM topology is called *survivable mapping* [2]. If the mapping stays connected after the failure of a single physical link, then such a mapping is declared a *link-survivable mapping*. Similarly, if the mapping remains connected after a single node failure, then it is called a *node-survivable mapping* [2].

As an example consider Fig. 1. The logical topology consists of a ring 1-2-4-6 as shown in Fig. 1(a). The physical topology is shown in Fig. 1(b). Fig. 1(c) shows a possible mapping of the logical topology. Here logical links (1,2), (2,4), (4,6) and (6,1) are mapped respectively to the physical paths 1-2, 2-3-4, 4-5-6 and 6-1. This routing is survivable, that is, any physical link failure does not disconnect the logical topology because these paths are link disjoint. For example, if the physical link 2-3 fails then the logical link 2-4 fails but the topology remains connected.

Figure 1(d), shows another possible mapping for the same logical and physical topologies. Here the logical links (1,2), (2,4), (4,6) and (6,1) are mapped respectively to the physical paths 1-2, 2-5-4, 4-5-6 and 6-1. However, this mapping is not survivable. Because if the physical link 4-5 fails, then the logical links 2-4 and 4-6 also fail, isolating node 4.

The above example illustrates the facts that multiple mappings may exist for a logical-physical topology pair and some of them may not be survivable or in some cases survivable mapping may not be possible. Determining whether a survivable mapping for a particular logical topology exists may require some degree of exhaustive search. In fact, the problem of finding a link survivable mapping has been widely studied and is proved to be NP-complete by Modiano et al [4]. NP-completeness requires exhaustive search approach that is unattractive. Therefore, a number of approximate algorithms have been proposed in the literature. The proposed algorithms fall under two categories, Integer Linear Programming (ILP) and heuristics [2]. ILP based algorithms have high time complexity even for networks containing only a few nodes. Heuristics employ *Tabu Search*, *Simulated Annealing*, or various algorithms using some form of shortest path algorithm. Such heuristics, however, may fail without providing any information about the existence of a survivable mapping. This problem bears some similarity to the widely studied problem of rerouting traffic carried by a failed link in ATM networks [5].

The objective of this paper is to study a recent heuristic presented in [2] by Thiran and Kurant to obtain survivable mappings for logical networks and to introduce new ones, that are able to map a higher percentage of logical topologies and are able to tolerate multiple faults in the physical network. Section II provides a brief overview of existing approaches in literature pertaining to this problem. Section III contains a detailed discussion of Thiran-Kurant approach [2], points out certain attractive features and difficulties with this approach. Section IV presents the enhancements to their basic approach to achieve better fault coverage and add robustness to the survivable routing schemes. In Section V we provide simulation results evaluating the new heuristics.

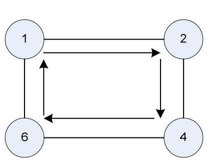


Fig. 1(a) Logical topology.

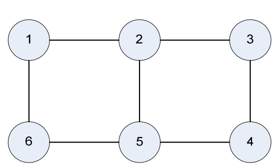


Fig. 1(b) Physical topology.

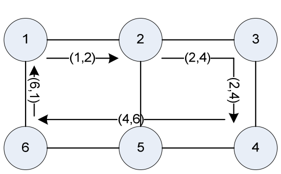


Fig. 1(c) A survivable mapping.

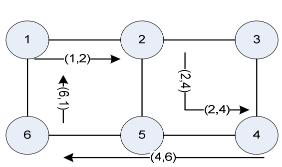


Fig. 1(d) An unsurvivable mapping.

II. RELATED WORK

The problem of designing survivable IP-over-WDM networks has been well studied. There have been a number of publications, which provide various approaches to design approximate solutions due to the NP-completeness nature of the problem of survivable mapping. The primary assumption in these works is that both logical and physical topologies have some degree of redundancy. Most of them assume that the topologies are at least two-connected and that there is no articulation point. These are also the assumptions of the research work done in this paper.

The concept of protection interoperable design was first introduced in [6]. A group of logical links that can support up to *k* logical link failures is called a protected group with protection level *k*. A protected group with protection level *k* is assumed to have the capability to reroute traffic along broken links of the group to some other links in the group as long as the number of broken links is no more than *k*. There is no guarantee of protection, if this condition is violated. The authors of [6] considered three scenarios. The bottleneck condition requires that a single failure in the physical network does not cause more than *k* logical links to fail within the protected group of protection level *k*. The connectivity condition requires that the logical topology remains connected for any single failure in the physical topology. This is the problem considered in our work in this paper. The third scenario allows more than one protected group. The authors of [6] emphasize that protection deployed by the higher layer must be taken into consideration in any logical topology design approach. The paper discusses an ILP formulation for the problem, which takes care of the three scenarios and also presents a heuristic called PIW algorithm that uses tabu search. The PIW algorithm can handle different cases of higher level protection strategies (SONET, IP, ATM etc.). The work in [6] also provides a general framework for protection interoperable design and offers several opportunities for generalizations and refinements. Ref [7] is a follow up of [6] and presents an approach that considers capacity constraints.

Ref. [4] takes Integer Linear Programming (ILP) approach to solve the problem for ring logical topologies. And proposes an ILP formulation based on the observation that a single physical link failure can disconnect the logical topology only if it carries the entire cutest of the logical topology. Since solving ILP formulations for large networks results in excessive computation times, several relaxations to the ILP formulation are proposed in [4] Ref [8] improves on [4] by providing MILP (Mixed ILP) formulations.

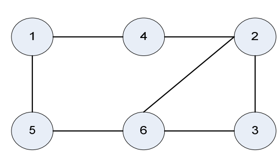


Fig. 2 (a) Logical Topology

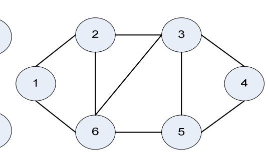
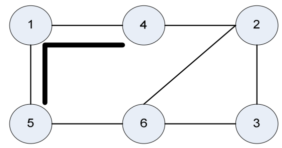


Fig. 2 (b) Physical Topology



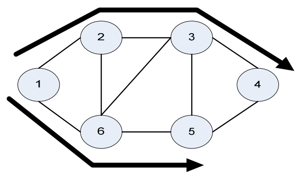
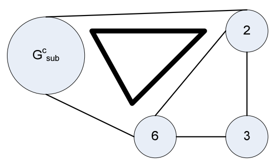


Fig. 2 (c) Logical Topology Fig. 2 (d) Physical Topology



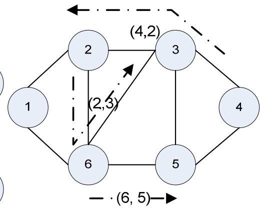
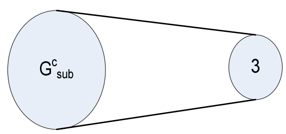


Fig. 2 (e) Logical Topology Fig. 2 (f) Physical Topology



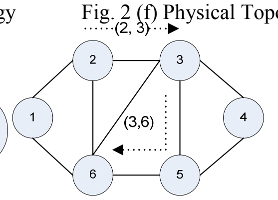


Fig. 2 (g) Logical Topology Fig. 2 (h) Physical Topology

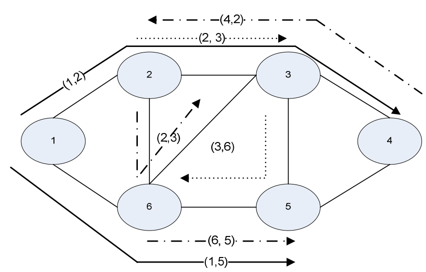


Fig. 2 (i) Final mapping

TABLE I
SMART ALGORITHM

Step 1: Start from the full logical topology.
Step 2: Pick a subgraph in the logical topology, and find a mapping for the links in the subgraph such that the mapping is link/node-survivable. IF no such (subgraph, mapping) pair is found,
THEN RETURN THE MAPPINGS THAT HAVE BEEN OBTAINED SO FAR AND THE REMAING CONTRACTED TOPOLOGY, END
Step 3: Update the mapping by adding this mapping to the set that contains previous successful mappings.
Step 4: Contract the logical topology by collapsing the edges included in the subgraph and merging the nodes, to create a single node.
Step 5: IF the contracted logical topology is reduced to a single node, THEN RETURN MAPPINGS OBTAINED,
END.
Step 6: GOTO Step 2, REPEAT THE STEPS FOR THE CONTRACTED TOPOLOGY.

Ref. [9] considers designing survivable mapping for logical topologies in the form of rings, in arbitrary physical topologies. It observes that a survivable mapping for a logical ring exits if the physical topology contains a closed trail visiting each node at least once. The paper provides a heuristic to find such a trail but does not consider capacity constraints.

In a more recent paper [2] Kurant and Thiran introduce the unique concept of *piecewise survivable mapping*, which involves recursively finding survivable mappings for small pieces (subgraphs) of the logical topology. The proposed algorithm is called SMART (Survivable Mapping Algorithm by Ring Trimming) and it is claimed that "*SMART never misses a solution if there is one.*" As we show in the next section this claim is not true unless certain connectivity requirement is satisfied. If SMART does not find a survivable mapping for the entire logical topology, then it returns a survivable mapping for a subset of the logical links and a set of unmapped links. This approach greatly simplifies the process of finding survivable mappings for fairly large networks. The solution is applicable to arbitrary logical and physical topologies. This algorithm can be applied to find link survivable mapping as well as to find node survivable mappings. This approach is discussed in detail in section III.

III. SMART ALGORITHM: AN ANALYSIS

Suppose the given logical topology is 2-edge connected or 2-veretex connected. SMART algorithm is applicable to both 2-edge connected and vertex connected cases. For the sake of simplicity in presentation, we restrict our discussion to 2-edge connected case. For detailed discussion of edge and vertex connectivity results reference [10] may be consulted. SMART starts by picking a *connected subgraph* of the logical topology (IP) and finding a mapping for it in the physical topology, which is survivable. If such a mapping is found, then the subgraph is contracted by collapsing the nodes and links of the subgraph in the logical topology. The process is repeated as long as a (subgraph, survivable mapping) pair in the contracted logical topology can be found. The algorithm terminates, if no such pair can be found. If after one or more successful applications of SMART, the logical topology is reduced to a single node, then a survivable mapping of the logical topology in the physical topology exists. The algorithm as described in [2] is shown in Table I.

SMART, in step 2 requires a subgraph which is connected. However, as we show in Fig 2 this requirement is not sufficient to guarantee survivability. This shortcoming may allow the SMART to terminate successfully but the resulting mapping may not be survivable. In Fig 2(c), SMART first selects a connected component consisting of links (5,1) and (1,4), which can mapped in survivable manner (Fig 2(c), 2(d)). In the next step, a subgraph consisting of links (4, 2), (2, 6), and (6, 5) is picked. This subgraph can be mapped in survivable manner as shown in Fig 2(e), and 2(f). Figures 2(g) and 2(h) show the mapping in the next step. The final mapping is shown in Fig 2(i). It can be seen that this mapping is not survivable. If the physical link (5, 6) fails then logical links (1, 5) and (5, 6) fail and node 5 is isolated.

The above example stresses that it is important to pick the subgraph in step 2 carefully. We can easily prove the following which is the corrected form of Theorem 2 in [2].

Theorem 1*: A 2-edge connected logical graph is survivable if and only if SMART algorithm picks in step 2 a 2-edge connected subgraph and terminates successfully.*

Proof: Necessity: SMART terminates successfully→ Graph is survivable. This follows from Theorem 1 in [2].

Sufficiency: SMART terminates unsuccessfully→Graph is not survivable. Proof is by contradiction. Assume that the SMART algorithm terminates unsuccessfully and the logical topology G is survivable. That is, at some iteration step 2 does not find a survivable 2-edge connected subgraph of the contracted graph G_c. Since G is survivable removing a physical link will result in G' that is connected. If we now contract those edges in G' corresponding to the edges defined by the contracted node in G', then the resulting graph G_c' will be connected. This means that removal of a physical link does not disconnect G_c. So G_c is survivable and SMART would terminate successfully at this step. A contradiction

Note that the above theorem holds even if the subgraph chosen in step 2 of SMART is not connected as long as every connected component of the selected subgraph is 2-edge connected

In SMART, one may be required to consider a large number of subgraphs in step 2 until a survivable subgraph is found. This is a computationally expensive task. So an implementation of SMART will be limited to considering a small number of subgraphs, and may terminate unsuccessfully, if none of these selected subgraphs is found to be survivable. This does not mean that survivable mapping of the entire logical topology does not exist. As an example consider Fig. 3. Assume that all the links in the subgraph {1,2,5} cannot be mapped disjointly. Also assume that links 1-5 and 2-5 share a physical links. However, if we are able to map all the remaining links in a disjoint manner then the logical topology will still remain connected. This leads us to the following:

Fact 1*:* If a survivable mapping cannot be found for a particular subgraph then accepting an unsurvivable mapping for this subgraph and continuing with SMART may still yield a survivable mapping for the entire logical topology.

The requirement that a 2-edge connected graph is selected can easily be met by choosing a cycle. The question then is whether we should select a longer or shorter cycle. A short cycle can be picked using a shortest path algorithm

Fact 2*:* A shorter cycle has an additional advantage over a longer cycle in the sense that the chance of successfully mapping a small number of links is higher than mapping a larger number of links. However, a longer cycle has the advantage of having one or more number of straddling links [11] that can be mapped in an arbitrary manner.

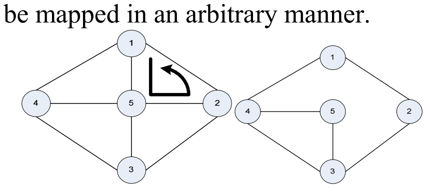


Fig. 3 (a) Logical topology

Fig. 3 (b) Logical topology after failure

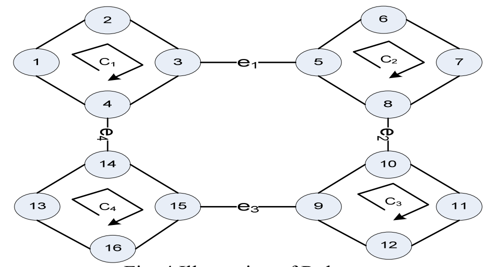


Fig. 4 Illustration of Robustness.

Finally, the problem of finding disjoint paths between different pairs of vertices is known to be NP-complete [12]. So, the implementation in [2] employs a simple heuristic to find link disjoint paths. Each link in the physical network is assigned a weight of 1, and an attempt is made to map the links of the subgraph using a shortest path algorithm. If each physical link is used only once, then a disjoint mapping is possible. If such a mapping cannot be found, then the weight of the links used more than once is increased by 1 and the process is repeated. The heuristic ends after a certain number of unsuccessful repetitions. The unsuccessful termination of the heuristic, however, does not imply that disjoint paths do not exist for the selected subgraph. So enhancement of this heuristic must investigated. This enhancement may involve searching a larger local space.

IV. MODIFICATIONS AND ENHANCEMENTS

SMART algorithm described in section III provides a unique and novel approach to find survivable mappings for logical topologies in physical topologies. Section III also discusses some of the issues involved in an efficient implementation of this algorithm. The objective of this section is to enhance the abilities of the SMART algorithm. Since the problem under study is NP-complete, some degree of exhaustive search is necessary before terminating the algorithm after an unsuccessful iteration.

The SMART algorithm is based on the following important property. Suppose the given logical topology is 2-edge connected. If a 2-edge connected subgraph, G_1 is routable in a survivable manner and a subgraph G_2 of the contracted subgraph is also routable in a survivable manner, then the subgraph of the logical topology containing the links of G_1 and G_2 is also routable in a survivable manner. Of course, here survivability is with respect to one link failure in the physical network. Note that it is enough to make sure that the subgraph selected for routing is minimal 2-edge connected. So the implementation of SMART given in [2] starts by picking a cycle which is minimal 2-edge connected. In the following, certain enhancements to the SMART that increase the probability of finding a survivable routing of the given logical topology and add certain additional features to the quality of the routing.

When the SMART algorithm terminates successfully, it gives a mapping which remains connected for any single physical link failure. This is true even if a single physical link failure causes more than one logical link to fail as long as these logical links are in different cycles constructed by the SMART algorithm. This leads us to definition of the concept of *robustness of a routing*. A survivable routing is considered *robust* if the logical topology remains connected for a large number of pairs of logical link failures. So, the larger the number of cycles considered by SMART the higher will be the robustness of the routing. For example, consider the logical topology in Fig. 4. If we select the cycles C_1, C_2, C_3, C_4 in this order and the cycle containing the edges e_1, e_2, e_3 and e_4 in the graph that results after contracting the four cycles, then the resulting survivable routing will remain connected for 160 pairs of logical link failures. On the other hand, if SMART first picks the cycle (3,5,8,10,9,15,14,4,3) and the cycles {(4,1),(1,2), (2,3)}, {(5,6), (6,7), (7,8)}, {(10,11), (11, 12), (12, 9)}, and {(15,16),(16,13),(13,14)}in the contracted graphs then the resulting logical topology will survive only 88 pairs of logical link failures. To achieve high robustness we propose the following enhancements to SMART.

Enhancement 1: When selecting a subgraph, effort should be made to not include contracted nodes as long as possible.

As pointed out in *fact* 1, even if a survivable mapping cannot be found for a particular subgraph then accepting this mapping and continuing with SMART may still yield a survivable mapping for the entire logical topology.

Enhancement 2: If a survivable mapping cannot be found for a particular subgraph then accept the current mapping and continue with SMART.

At the end of the SMART algorithm some of the logical links may be left unmapped. In fact those logical links can be mapped in an arbitrary manner. By a judicious mapping of these links one can achieve a high success rate for obtaining survivable mappings. This is similar to taking advantage of DON'T CARE conditions in logical function minimization. Furthermore, if a subgraph contains a contracted node, then only two of the multiple links incident on it are mapped disjointly. The remaining links are not mapped. However, mapping these links in an arbitrary manner (or disjoint manner, if possible) may allow some of the logical topologies to become survivable that were previously declared unsurvivable. So we propose the following enhancement to SMART.

Enhancement 3: At the end of SMART, map unmapped links in an appropriate manner to reduce the degree of unsurvivability of the final mapping.

The algorithm used in [2] to map a set of links, uses a variant of shortest path algorithm, and therefore does a local search. Each time the algorithm fails to find a disjoint mapping, it increments the weights of physical links used more than once by 1. Increments of 1 may require a large number of applications of shortest path algorithm before disjoint mappings are found. As an example, assume that the physical topology under consideration is shown in Fig 4. Also assume that SMART is trying to find a mapping between logical nodes 5 and 7, and the discovery of path 5-3-4-14-15-9-10-8-7 (length 8) is required so that the mapping for a given subgraph is survivable. The algorithm will alternate between paths 5-6-7 and 5-8-7 several times before choosing the correct path. However, penalizing the repeated links heavily, may force the algorithm to converge faster. As an alternative to incrementing by 1, each physical link is assigned a cost of 2^k, where k is the number of times the link appears in unsuccessful attempts. Initially, $k = 0$, however in the next iteration k is incremented by 1 and the links used more than once have cost 2. Similarly, in the next iteration if any of the links repeated in previous iteration appear again their cost is incremented to 4.

4) ***Enhancement 4:*** Before finding shortest paths assign each physical link a cost of 2^k, where k is the number of times the link appears in unsuccessful attempts.

V. SIMULATION STUDY

To study the behavior of SMART and the modifications proposed in section IV, they were implemented using LEDA [13] and VC++ 7.0. For simulation two kinds of physical topologies with a varying number of nodes and edges were generated. Regular topologies of n nodes were generated with an average degree of 6. The regular topologies were constructed using a procedure originally given by Harary and described in [10], and random topologies were generated using LEDA with n nodes and $m = 3 \times n$ edges. The values of n were {20,30,40,50}. Logical topologies were generated randomly with number of nodes being $f \times n$, where f = {0.5,0.6,0.7,0.8,0.9,1.0} and the number of edges were $1.5 \times (f \times n)$. The logical nodes were always a random subset of the physical nodes. For each f, 100 random logical topologies were generated. After generating the topologies, they were admitted for further processing only if they were at least two edge connected. Each logical-physical topology pair was subjected to following two methods:

Method 1(KT): The first method was an implementation of Kurant-Thiran approach presented in Table I. A subgraph was chosen randomly and then attempts were made to map it in disjoint manner. If a disjoint mapping could not be obtained after applying the mapping method 50 times, a new subgraph was selected and SMART terminated after unsuccessfully examining 25 subgraphs.

Method 2(M-KT): This method implemented Kurant-Thiran approach with enhancements 1, 2, 3, and 4, proposed in section IV. In the implementation; *a*) an attempt was made to find a subgraph without a contracted node; *b*) an unsurvivable mapping for a subgraph was accepted; *c*) all the links not previously mapped, were mapped; *d*) the cost function described in section IV was used. *e*) At the end of the execution, a test was conducted to see if the logical topology remained connected in case of a single physical link failure.

The statistics of interest were the percentage of mapped logical topologies and robustness. Methods 1 and 2 were applied on the same logical-physical pairs, and tests were conducted with n = {20,30,40,50}. However, the results are shown for $n = \{20,50\}$ only due to space limitations. Fig. 5 and 5 show the ratio of survivable topologies found to the number of logical topologies considered. Fig. 5 and Fig. 6 compare both the methods for $n = 20$ and 50, respectively. It can be seen that both methods are comparable when f is small, but for a higher value of f, method 2 performs significantly better than method 1. Fig 6 shows results for $n = 50$. Also, as $f \rightarrow 1$, the logical topology grows denser and the percentage of mapped topologies decreases as more number of physical links are likely to be common to several paths (logical links).

Fig. 7 and 8 show comparison of robustness for $n = \{20,50\}$. It can be seen that both methods provide a high degree of robustness. However, M-KT performs slightly better in each case as explained in Section IV.

CONCLUSION

In this paper the problem of routing logical topologies onto physical topologies in WDM optical networks is considered. We have given an analysis of a recent novel algorithm called SMART due to Kurant and Thiran [2], and pointed out certain attractive features of this algorithm. We have also shown that the claim made in [2] that SMART, on termination, will determine if a given logical topology admits survivable routing, is not correct. We have given in Theorem 1 a corrected form of Theorem 2 in [2]. The concept of robustness of a routing is defined. We have suggested several enhancements to SMART to make it more effective and produce routings that are robust. Simulation results are given to show the effectiveness of the new heuristics.

ACKNOWLEDGEMENT

The work of K. Thulasiraman has been supported in part by NSF ITR grants ANI-0312435 and ECS 04-26831 and the work of Xue has been supported by NSF ITR grant ANI-0312635.

REFERENCES

[1] G. Iannaccone, C.-N. Chuah, S. Bhattacharyya, and C. Diot, “Feasibility of IP restoration in a tier-1 backbone,” *Sprint ATL Research Report Nr. RR03-ATL-030666*.

[2] M. Kurant, and P. Thiran, “On Survivable Routing of Mesh Topologies in IP-over-WDM Networks,” *Proc. of IEEE INFOCOM’ 05,* Vol. 2, pp. 1106-1116, 2005.

[3] I. Chlamtac, A. Ganz, and G. Karmi, “Lightpath communications: An approach to high-bandwidth optical WAN’s,” *IEEE Transactions on Communications,* Vol. 40, No. 7, pp. 1171-1182, 1992.

[4] E. Modiano and 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, pp. 800-809, 2002.

[5] K. Murakami, H. S. Kim, “Comparative Study on Restoration Schemes of Survivable ATM Networks,” *Proc. of IEEE INFOCOM '97,* Vol. 1, pp. 345-352, 1997.

[6] O. Crochat, J.Y.Le Boudec, “Design Protection for WDM Optical Networks”, *IEEE Journal on Selected Areas in Communication*, Vol. 16, No. 7, pp. 1158-1165, 1998.

[7] J O.Crochat, J.Y.Le Boudec, O.Gerstel “Protection Interoperability for WDM Optical Networks”, *IEEE Transaction on Networking*, Vol. 8, No. 3, pp. 384-395, 2000.

[8] Qi Deng, Galen Sasaki, and Ching-Fong Su, “Survivable IP Over WDM: An Efficient Mathematical Programming Problem Formulation” *Proceedings of the 40th Annual Allerton Conference on Control, Communication, and Computing*, Oct. 2002.

[9] A. Sen, B. Hao, B. H. Shen and G. Lin, “Survivable Routing in WDM Networks – Logical Ring in Arbitrary Physical Topology,” *Proceedings of IEEE International Conference on Communications*, pp. 2771-2775, 2002.

[10] K. Thulasiraman and M. N. S. Swamy, *GRAPHS: Theory and Algorithms*, Wiley-Inter-science 1992.

[11] W. D. Grover and D. Stamatelakis, "Cycle-oriented Distributed Preconfiguration: Ring-like Speed with Mesh-like Capacity for Self-Planning Network Reconfiguration", *Proceedings of IEEE International Conference on Communications,* pp. 537–543, 1998.

[12] A. Frank, *Packing paths, circuits and cuts – a survey* (*in Paths, Flows and VLSI-Layout*), Springer, Berlin, 1990.

[13] K. Mehlhorn and S. Näher, *LEDA: A Platform for Combinatorial and Geometric Computing*, Cambridge University 1999.

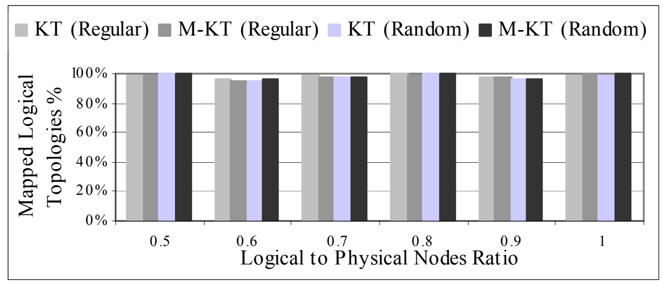


Fig. 5 Percentage of Mapped Logical Topologies. $n = 20$

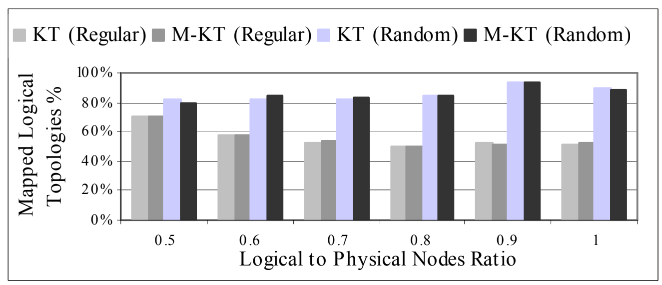


Fig. 6 Percentage of Mapped Logical Topologies. $n = 50$

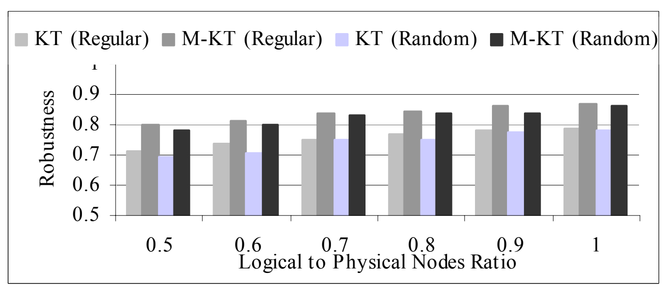


Fig. 7 Comparison of Robustness. $n = 20$

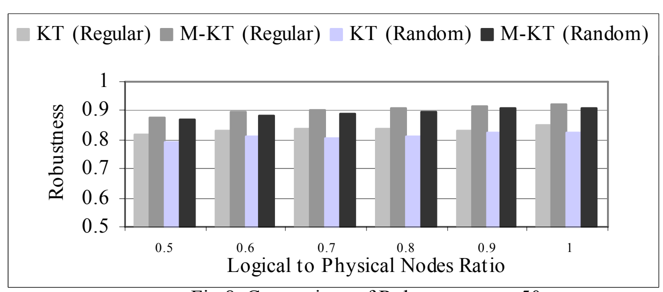


Fig 8. Comparison of Robustness. $n = 50$