Fault Detection and Path Performance Monitoring in Meshed All-Optical Networks

Hongqing Zeng, *student member, IEEE,* and Changcheng Huang, *member, IEEE*
Advanced Optical Network Laboratory, Department of Systems and Computer Engineering,
Carleton University, 1125 Colonel By Drive, Ottawa, ON, Canada K1S 5B6

Alex Vukovic and J. Michel Savoie
Communications Research Centre Canada, 3701 Carling Ave., Ottawa, ON, Canada K2H 8S2

Abstract– **Fault detection is critical for all-optical networks (AONs). This paper introduces the concept of monitoring cycles and proposes a mechanism for fault detection and path performance monitoring based on decomposing AONs into monitoring cycles. Two monitoring cycle finding algorithms are developed: heuristic depth first searching (HDFS) and shortest path Eulerian matching (SPEM). The two algorithms are compared in terms of wavelength overhead in nodes and links. The comparison results are obtained for four typical networks, including NSFNET, ARPA2, SmallNet and Bellcore. It is shown that the proposed fault detection mechanism based on monitoring cycles is effective and cost efficient.**

Keywords*– *All-optical networks, fault detection, cycle cover, monitoring cycles

I. INTRODUCTION

With the development and deployment of dense wavelength division multiplexing (DWDM) and all-optical networking technologies, telecommunication transmission networks continue to evolve towards higher data rates and increased wavelength numbers and density. It greatly improves the data transmission efficiency but, at the same time, even a very short disruption of service caused by a network fault may lead to a very high data loss in such networks. Consequently the network function for monitoring and fault detection is critical for such networks.

There are numerous fault detection mechanisms for traditional electrical communication networks. Unfortunately, such mechanisms cannot be applied directly to all-optical networks (AONs) due to the lack of electrical terminations in AONs. Even some detection methods deployed in optical networks with opto-electro-opto (OEO) conversion cannot be adapted to AONs. Reference [1], for example, has shown that the schemes in SDH/SONET could not be applied to AONs.

Optical power detection, optical spectral analysis, pilot tone and optical time domain reflectometry can be deployed for fault detection and are also applicable to attack detection [2] in AONs. Reference [3] developed a fault detection scheme by assigning monitors to the sinks of each optical multiplex section and optical transmission section. A heuristic algorithm was proposed in [4] to efficiently assign monitors and thus reduce the required number of monitors. This kind of scheme is channel-based and introduces large numbers of monitors thus it is not feasible in current AONs due to channel dynamics, scalability and costs factors.

Other methods, e.g. a finite state machine described in [5], were proposed but their complexity for large-scale and dynamic networks impedes their deployment.

Most routing protocols, e.g. OSPF and IS-IS, also have inherent fault detection functionality. Some key OSPF parameters were optimized in [6] to achieve fast fault detection. A joint optical and IP layer method was proposed in [7] to accelerate the detection speed. Usually the fault detection time of routing protocols is in the seconds range, even with some accelerating techniques. However, the typical time constraint for fault recovery in optical networks is 50 milliseconds. This constraint inhibits moving the fault detection from the optical layer to the IP layer. Thus some effective and efficient fault detection mechanisms at optical layer are still expected.

In this paper, we propose a mechanism for fault detection and path performance monitoring in AONs that utilizes dedicated wavelengths as supervisory channels. The monitors are assigned based on cycle covers of the network topology. Two cycle cover finding algorithms are compared in terms of the cycle numbers and length, as well as the maximum and average number of occupied wavelengths in nodes and links, for four typical networks: NSFNET, ARPA2, SmallNet and Bellcore.

This paper is organized into the following sections. Section II introduces the concept of monitoring cycles and develops two cycle finding algorithms. The proposed scheme is applied to some network examples and the performances of the given algorithms are evaluated in Section III. Some conclusions are drawn and future work is outlined in Section IV.

II. MONITORING CYCLES

In AONs the fault impact scopes are various in terms of wavelengths (channels or paths). Some faults only affect a single or some specific wavelengths, e.g. transmitter laser failure, optical crossconnect (OXC) port blocking, etc. Others may affect all the wavelengths passing through the faulty module, e.g. fibre cut, optical amplifier (OA) saturation, etc. The characteristics of AON faults are strongly related to the specific network components [8]. For AONs, faults affecting specific wavelengths usually can be handled by in-place fault control or management entities. On the other hand, network faults affecting all wavelengths put an impact on more user traffic and generate much more alarms in the network, thus degrade the network performance more severely. Therefore it

0-7803-8794-5/04/$20.00 (C) 2004 IEEE

is critical to develop some mechanisms to address the latter kind of network faults.

In this section we describe a scheme for detecting the network node or link faults that affect all lightpaths passing through the node/link. The main idea is to decompose a network into cycles. All nodes and links of the given network are covered at least by one cycle. We define these cycles as "***monitoring cycles***". A pair of transmitter and receiver is assigned to one node in each monitoring cycle and thus a loopback dedicated supervisory channel (using an independent wavelength) is set up. Such a mechanism can achieve fault detection and path performance monitoring in AONs.

A. Feasibility

A network can be modelled as a finite undirected graph $G(V,E)$, where V is the set of vertices (network nodes) and E is the set of edges (network link). A *cycle* (denoted as *c*) of the graph G is a sub-graph of G, which is connected and regular of degree two. A cycle is often identified with its edge-set. A *cycle cover* (denoted as *C*) of a graph is a family of cycles in which each vertex and edge of the graph appears at least in one of these cycles. Figure 1 gives an example of a network and an instance of its cycle cover. This cover consists of 4 cycles. Some nodes and links appear only in one cycle of the cover. For example node *c*, link *bc* and *cg*, are covered by cycle *(4)* only. But some others appear in multiple cycles, e.g. edge *bd* is covered by both cycle *(1)* and *(4)*, node *f* by cycle *(2)*, *(3)* and *(4)*.

To achieve fault detection and path performance monitoring for all nodes and links of a given network, a cycle cover has to be found. At the same time, for maximizing the network resource utilization, we also have to minimize the number of wavelengths occupied by monitoring channels in nodes and links. That is to say, the goal is to find a cycle cover *C* for graph $G(V,E)$ that minimizes the number of each node and link's occurrence in all monitoring cycles. *C* will be called a *cycle double cover* if every edge appears in exactly two of those cycles of *C*. The following conjecture was studied in [9],

Cycle Double Cover Conjecture: Every bridgeless graph has a cycle double cover.

Although the conjecture has not been completely proven, it was shown in [9] that a minimum counterexample to the cycle

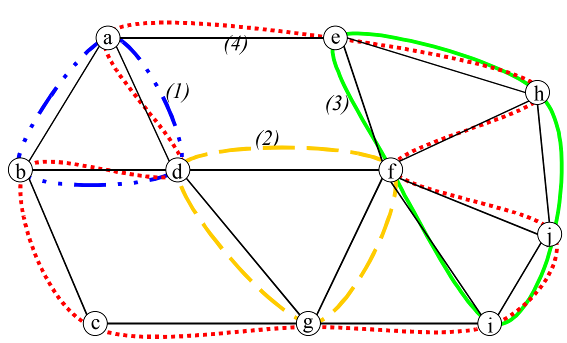


Fig. 1. A graph example and its cycle cover

double cover conjecture must be a snark that has girth at least seven. A snark is a cyclically 4-edge-connected cubic graph of girth at least five. It is worth to note that no snark of girth at least seven is known. In fact, some literatures, e.g. [10], had conjectured that such snarks did not exist. Thus it is safe to say, even if the counterexamples to the conjecture do exist, it is not expected that communication networks with such topologies will be encountered in the real world.

The cycle double cover conjecture not only shows the feasibility of the setup of monitoring cycles, but also gives a reference for evaluating the performance of monitoring cycle finding algorithms in terms of the network resource utilization. That is, an achievable reference of the average number of occupied wavelengths is two for monitoring in all links.

In the following sub-sections we describe two heuristic algorithms for finding a cycle cover in given graphs: the heuristic depth first searching algorithm and the shortest path Eulerian matching algorithm.

B. Heuristic depth first searching (HDFS)

Given a graph $G(V,E)$, starting from any node $n \in V$, we can traverse all links in E by depth first searching (DFS). Let the traversed part be $G'(V',E')$ during the DFS. While a link e from node x to node y being traversed, if $y \in V'$, then there must exist a path $p(y,\cdots,x) \in G'$. Thus path $p(y,\cdots,x)$ and link $e(x,y)$ form a cycle. Based on this fact, a heuristic cycle cover finding algorithm is developed as below,

1) Given graph $G(V,E)$, let the cycle cover $C = null$; number all nodes in *V*; and label all nodes in *V* and all links in *E* as "uncovered";
2) Select an uncovered link *e* in E, if multiple such links are available, select the uncovered link whose endpoints are also uncovered. Start DFS from *e* and go to that uncovered endpoint of *e* if possible; if no uncovered link with uncovered endpoint is available, apply the largest/smallest rule described below;
3) At each step of the DFS, select an uncovered link if possible. If multiple links are available, alternatively use the largest/smallest node number first rule in the iteration, e.g. if the last time we selected the node with the largest number among multiple nodes with the same priority, then this time we select the node with the smallest number;
4) Once a link returns to the previously visited part, a cycle *c* can be formed and add the cycle to the cover *C*; label all the links and nodes in cycle *c* as "covered";
5) Repeat (2)-(4) until all links in *E* are "covered".

The choice of starting link tries to avoid covering a link with many different cycles in the cover. The alternative largest/smallest numbered node first rule distributes cycles evenly among nodes and links. Both heuristic rules therefore avoid occupying large number of wavelengths for monitoring.

Fig. 2 describes the procedure of HDFS applied to the graph example given in Fig. 1. Before starting the DFS, the nodes are numbered from 1 to 10. All nodes and links are labeled as "uncovered" and set $C = null$. During the DFS the following iterations are executed,

Iteration 1, start from node 1 and there are 3 "uncovered" links: *(1,2), (1,4) and (1,5)* . Applying the smallest numbered node first rule, we select link *(1,2)* . After the DFS the cycle *1-2-3-7-4-1* is obtained and added to *C*. Nodes in set *{1,2,3,7,4}* and links in set *{(1,2), (2,3), (3,7), (7,4), (4,1)}* are labeled as "covered".

Iteration 2, start from node 5 and there are 3 "uncovered" links: *(5,1), (5,6) and (5,8)* . Since node *6* and *8* are uncovered, links *(5,6)* and *(5,8)* are prior to *(5,1)* . Alternatively to iteration 1, we apply the largest numbered node first rule and select link *(5,8)* . After the DFS the cycle *5-8-10-9-6-5* is obtained and added to *C*. Nodes in set *{5,8,10,9,6}* and links in set *{(5,8), (8,10), (10,9), (9,6), (6,5)}* are labeled as "covered".

Similarly, cycles *6-4-2-1-5-6, 8-6-10-8,* and *6-7-9-6* are obtained and added to *C* in iterations 3, 4, 5 respectively. After iteration 5 all links in the graph are covered and a 5-cycle cover is obtained as shown in Fig. 3.

C. Shortest path Eulerian matching (SPEM)

For an Eulerian graph, there exists an Eulerian cycle that covers all links once. If we traverse the Eulerian cycle by following links in it until a node is re-visited, the traversed part forms a cycle. Then we remove this part from the Eulerian cycle and traverse the remaining part until all links are removed. In this way, the Eulerian cycle is decomposed into a cycle cover *C*. Due to the fact that no two cycles in *C* have a common link; the minimum number of monitoring wavelengths incident to each link can be achieved.

Euler proved that a graph is Eulerian if and only if every node has an even degree. Thus a non-Eulerian graph has some nodes with odd degrees. Since each link connects two nodes, the total number of odd-degree nodes is even. We can augment the given graph to construct an Eulerian graph by adding links between pairs of odd-degree nodes, i.e. Eulerian matching. In the matching process each added new link corresponds to a path consisting of existing links between the node pair in the original graph. Links included in one augmentation will be covered one more time. To minimize the average number of wavelengths occupied by monitoring in links, i.e. minimize the average link cover times, the shortest-path augmentations between odd-degree node pairs are added. This heuristic shortest path Eulerian matching (SPEM) is described below,

(1) For a non-Eulerian graph $G(V, E)$, find the set V' of odd-degree nodes;

(2) Start from a node $x \in V'$ and find the shortest path to every other node, select the smallest one among them, denote as $p(x, y)$. Add path $p(x, y)$ to G (now some links in G are "doubled") and remove x, y from V' ;

(3) Repeat (2) until $V' = null$. Now $G(V, E)$ is Eulerian;

(4) Find an Eulerian cycle of the augmented $G(V, E)$ and decompose it to a cycle cover as mentioned above.

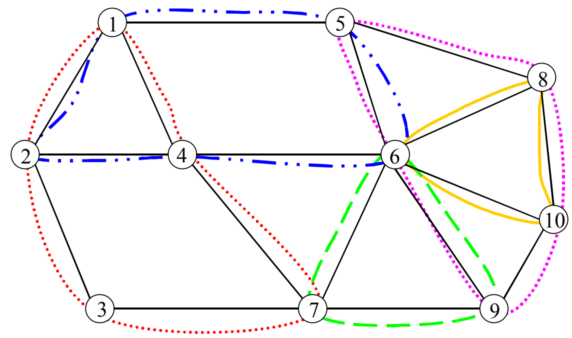


Fig. 2. Finding cycle cover by heuristic depth first searching

For example, in the graph given in Fig. 1 we firstly label the degree for all nodes and the odd-degree node set is *{1,2,5,8,9,10}*. For node 1, the shortest paths to another node are *(1,5)* and *(1,2)*, which are single hops. Select *(1,2)* and remove nodes *1,2* from the odd-degree node set. Repeatedly we get the matching path set {*1-2*, *5-8*, *9-10*} (total length is 3) as shown in Fig. 3. If we select (1,5) at the first step, the matching path set would be {*1-5*, *2-3-7-9*, *8-10*}. The total length is 5, larger than the first matching path set and thus was dropped. In this way by enumerating all possible tiers at each step, we can get the shortest path matching. The node degrees changed by the matching paths are labelled in the brackets. Now all node degrees are even and the augmented graph is Eulerian.

An Eulerian cycle can be found by any existing traversing algorithms, such as DFS. In this example an Eulerian cycle is,

1-5-8-10-9-10-6-9-7-6-8-5-6-4-7-3-2-4-1-2-1

This Eulerian cycle is decomposed into 4 cycles as shown in Fig. 4. Note that a two-edge cycle, e.g. *10-9-10*, is not considered as a "real" cycle.

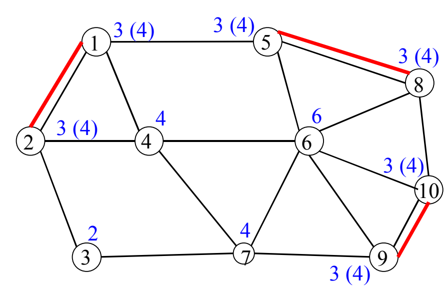


Fig. 3. Shortest path Eulerian matching

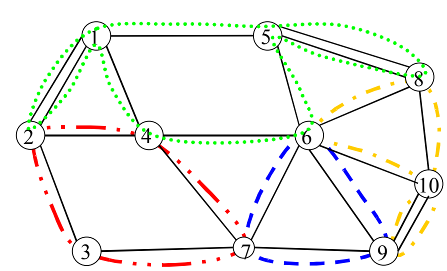


Fig. 4. Cycle cover obtained by SPEM

0-7803-8794-5/04/$20.00 (C) 2004 IEEE

III. ALGORITHMS EVALUATION

We tested and compared the two heuristic algorithms on four typical networks, including NSFNET, ARPA2, SmallNet and Bellcore. Their topologies and cycle covers obtained by HDFS and SPEM are shown in Figures 5-8 respectively. The performance of the cycle cover finding algorithms are evaluated using the following metrics,

Cycle number is the number of cycles in a cover instance.

Cycle length is the number of links in a single cycle.

Maximum cycle length is the maximum length of a single cycle in the cover.

Average cycle length is the average length of all cycles.

Maximum number of occupied wavelengths in a node/link is the maximum number of wavelengths assigned for monitoring in a single node/link. It is equivalent to the maximum cover times of a single node/link in a cycle cover.

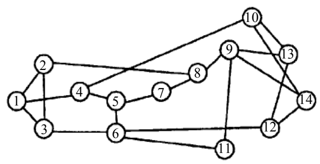
Average number of occupied wavelengths is the average number of wavelengths assigned for monitoring in all nodes or links. It is equivalent to the average node/link cover times.

These are calculated for 4 typical networks using HDFS and SPEM respectively. The results are listed in Table 1.

The cycle number represents the total number of required transmitters and receivers, as well as wavelengths for monitoring, thus it is the measurement of the cost. The results in Table 1 show that the cycle numbers for the networks are pretty small relative to the number of their nodes and links. Therefore the proposed mechanism is a cost efficient method for fault detection and path performance monitoring in AONs.

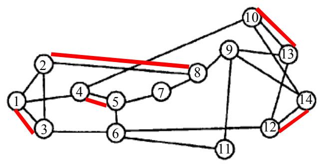
The cycle length implies the fault localization capabilities. Obviously, it is easier to precisely isolate and localize the network faults in shorter cycles.

Since each monitoring cycle occupies a dedicated wavelength along all nodes and links in this cycle, we must minimize the maximum and average number of occupied wavelengths, i.e. minimize the cover times of nodes and links in a cycle cover. The results in Table 1 have shown that the numbers of occupied wavelengths in all networks are small:



NSFNET: 14 nodes, 21 links

Cycle cover obtained by HDFS:
Cycle 1: 10-14-12-13-10
Cycle 2: 1-2-3-1
Cycle 3: 5-7-8-9-11-6-5
Cycle 4: 4-1-2-8-7-5-4
Cycle 5: 9-14-12-6-3-2-8-9
Cycle 6: 4-10-13-9-8-2-1-4

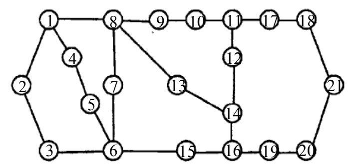


Shortest path Eulerian matching

Cycle cover obtained by SPEM:
Cycle 1: 1-4-5-4-10-13-9-8-2-1-3-1
Cycle 2: 10-14-12-13-10
Cycle 3: 9-14-12-6-11-9
Cycle 4: 8-7-5-6-3-2-8

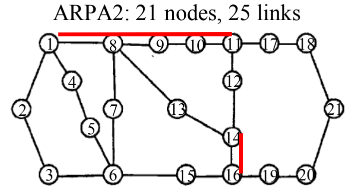
Eulerain cycle: *1-4-5-4-10-14-12-13-10-13-9-14-12-6-11-9-8-7-5-6-3-2-8-2-1-3-1*

Fig. 5. NSFNET: cycle covers obtained by HDFS and SPEM



ARPA2: 21 nodes, 25 links

Cycle cover obtained by HDFS:
Cycle 1:16-19-20-21-18-17-11-12-14-16
Cycle 2: 2-1-4-5-6-3-2
Cycle 3: 8-9-10-11-12-14-16-15-6-7-8
Cycle 4: 1-8-13-14-16-15-6-3-2-1

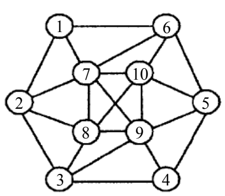


Shortest path Eulerian matching

Cycle cover obtained by SPEM:
Cycle 1: 1-8-13-14-16-15-6-7-8-1
Cycle 2: 14-16-19-20-21-18-17-11-12-14
Cycle 3: 8-9-10-11-10-9-8
Cycle 4: 1-4-5-6-3-2-1

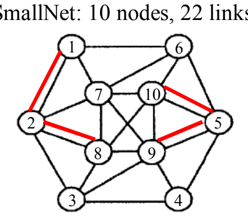
Eulerain cycle: *1-8-13-14-16-19-20-21-18-17-11-12-14-16-15-6-7-8-9-10-11-10-9-8-1-4-5-6-3-2-1*

Fig. 6. ARPA2: cycle covers obtained by HDFS and SPEM



SmallNet: 10 nodes, 22 links

Cycle cover obtained by HDFS:
Cycle 1: 10-9-5-6-10
Cycle 2: 1-2-3-4-5-10-7-1
Cycle 3: 8-10-9-8
Cycle 4: 6-1-2-7-6
Cycle 5: 9-4-5-10-7-9
Cycle 6: 9-4-3-8-9
Cycle 7: 7-8-2-1-7
Cycle 8: 3-9-4-3

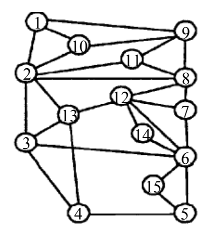


Shortest path Eulerian matching

Cycle cover obtained by SPEM:
Cycle 1: 7-10-9-8-7
Cycle 2: 9-5-10-6-7-2-3-9
Cycle 3: 5-6-1-2-8-10-5
Cycle 4: 1-7-9-4-5-4-3-8-2-1

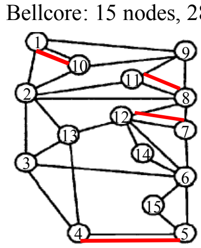
Eulerain cycle: *1-7-10-9-8-7-9-5-10-6-7-2-3-9-4-5-6-1-2-8-10-5-4-3-8-2-1*

Fig. 7. SmallNet: cycle covers obtained by HDFS and SPEM



Bellcore: 15 nodes, 28 links

Cycle cover obtained by HDFS:
Cycle 1: 12-14-6-15-5-4-13-12
Cycle 2: 2-3-4-5-6-7-8-2
Cycle 3: 11-8-12-7-6-3-13-2-11
Cycle 4: 1-2-10-9-1
Cycle 5: 9-11-8-9
Cycle 6: 1-10-2-3-4-5-6-12-7-8-9-1



Shortest path Eulerian matching

Cycle cover obtained by SPEM:
Cycle 1:1-10-9-11-8-2-1
Cycle 2:8-12-13-2-3-6-7-12-7-8
Cycle 3:12-14-6-15-5-6-12
Cycle 4:13-4-5-4-3-13
Cycle 5:2-11-8-9-1-10-2

Eulerain cycle: *1-10-9-11-8-12-14-6-15-5-6-12-13-4-5-4-3-13-2-11-8-9-1-10-2-3-6-7-12-7-8-2-1*

Fig. 8. Bellcore: cycle covers obtained by HDFS and SPEM

TABLE 1. COMPARISON OF CYCLE COVER FINDING ALGORITHMS: HDFS AND SPEM

Network	**NSFNET**		**ARPA2**		**SmallNet**		**Bellcore**	
Algorithm	HDFS	SPEM	HDFS	SPEM	HDFS	SPEM	HDFS	SPEM
Number of nodes	14		21		10		15	
Number of links	21		25		22		28	
Number of cycles	6	4	4	4	8	9	6	5
Max cycle length	7	11	10	9	7	4	11	9
Avg. cycle length	5.50	6.50	8.50	7.50	4.25	6.50	6.67	6.40
Max # of λs per node	4	2	3	2	5	3	4	3
Avg # of λs per node	2.36	1.71	1.62	1.29	3.40	2.50	2.67	1.87
Max # of λs per link	3	2	3	2	3	2	3	2
Avg # of λs per link	1.57	1.24	1.36	1.20	1.55	1.18	1.43	1.14

maximum $5\ \lambda s / node$ and $3\ \lambda s / link$ using HDFS, and maximum $3\ \lambda s / node$ and $2\ \lambda s / link$ using SPEM. The wavelength overhead, *H*, brought to the network by monitoring cycles can be quantitatively evaluated by,

$$H = \frac{maximum/average\ \#\ of\ occupied\ \lambda s}{number\ of\ available\ \lambda s}\ (a\ node/link)$$

Nowadays along with the deployment of DWDM technology, the number of wavelengths in a single link tends to become larger and larger. For example, it is reported even in 2001 that 432 wavelengths could be multiplexed into a single fibre [8]. Under this condition, the maximum overhead is only *1.16% per node* and *0.7% per link* using HDFS, *0.7% per node* and *0.5% per link* using SPEM. Such overhead doesn't impact the network utilization much, if it is not negligible.

IV. CONCLUSIONS

This paper introduced the concept of monitoring cycles and proposed a fault detection and path performance monitoring mechanism based on decomposing AONs into monitoring cycles. The heuristic depth first searching (HDFS) and shortest path Eulerian matching (SPEM) algorithms are developed for finding monitoring cycles in AONs. The two algorithms are compared with respect to the maximum and average number of wavelengths occupied by monitoring in nodes and links. The results for the 4 network examples show that the wavelength overhead is pretty low with this mechanism. Thus the proposed mechanism based on monitoring cycles is a promising fault detection method for AONs. It is also applicable to path transmission performance monitoring. The results also suggest that the SPEM algorithm is better than the HDFS algorithm in terms of the wavelength overhead.

Future work might include the development of more effective and efficient algorithms for finding cycle covers to further reduce the wavelengths overhead in links and nodes. The integration of the fault detection mechanism with the control plane is also one of the future work targets.

ACKNOWLEDGMENT

The authors would like to acknowledge Dr. Jing Wu and Mr. Baohua Zhang for their stimulating discussions.

REFERENCES

[1] Y. Kobayashi, Y. Tada, S. Matsuoka, N. Hirayama, and K. Hagimoto, "Supervisory systems for all-optical network transmission systems," *IEEE Globecom'96,* pp. 933-937, 1996

[2] M. Médard, D. Marquis, and S. R. Chinn, "Attack detection methods for all-optical networks," *Network and Distributed System Security Symposium,* 1998

[3] Y. Hamazumi, M. Koga, K. Kawai, H. Ichino, K. Sato, "Optical path fault management in layered networks," *IEEE Globecom'98,* vol. 4, pp. 2309-2314, 8-12 Nov. 1998

[4] S. Stanic, S. Subramaniam, H. Choi, G. Sahin, and H.-A. Choi, "On monitoring transparent optical networks," *Int'l Conf. on Parallel Processing Workshops,* pp. 217-223, Aug. 2002

[5] C.-S. Li, and R. Ramaswami, "Automatic fault detection, isolation, and recovery in transparent all-optical networks," *IEEE J. of Lightwave Tech.,* vol. 15, No. 10, pp. 1784-1793, Oct. 1997

[6] M. Goyal, K. K. Ramakrishnan, and W.-C. Feng, "Achieving faster failure detection in OSPF networks," *IEEE ICC'03*, 2003

[7] C. Assi, Y. Ye, A. Shami, S. Dixit, and M. Ali, "A hybrid distributed fault-management protocol for combating single-fiber failures in mesh-based DWDM optical networks," *IEEE Globecom'02*, 2002

[8] S. V. Kartalopoulos, *Fault Detectability in DWDM – Toward Higher Signal Quality & System Reliability*, Piscataway: IEEE Press, 2001

[9] F. Jarger, "A survey of the cycle double cover conjecture," in: *Cycles in Graphs*, Edited by B. R. Alspach and C. D. Godsil, Annals of discrete mathematics 27, New York: Elsevier Science, 1985

[10] F. Jarger, and T. Awart, "Conjecture 1," in: *Combinatorics 79*, Edited by M. Deza and I. G. Rosenberg, Annals of discrete mathematics 9, New York: Elsevier Science, 1980