

Monitoring Cycles for Fault Detection in Meshed All-Optical Networks

Hongqing Zeng and Changcheng Huang
Dept. of Systems and Computer Engineering
Carleton Univ., 1125 Colonel By Dr.
Ottawa, ON, Canada K1S 5B6
{hzeng, huang}@sce.carleton.ca

Alex Vukovic
Communications Research Centre Canada
3701 Carling Ave., P.O. Box 11490, Stn. H
Ottawa, ON, Canada K2H 8S2
alex.vukovic@crc.ca

Abstract

Fault detection is critical for all-optical networks (AONs). This paper introduces the concept of monitoring cycle and proposes a fault detection mechanism based on decomposing AONs into a set of cycles (a cycle cover), in which each one is defined as a monitoring cycle. Two cycle-finding algorithms are developed and compared for the proposed fault detection mechanism: heuristic depth first searching (HDFS) and shortest path Eulerian matching (SPEM). The degradation of wavelength utilization and the cardinality of cycle covers are analyzed for evaluating the proposed mechanism.

The proposed mechanism is applied to four network examples: NSFNET, ARPA2, SmallNet and Bellcore. The evaluation results show that the proposed fault detection mechanism is effective and cost efficient.

1. Introduction

With the development and deployment of dense wavelength division multiplexing (DWDM) technology, all-optical networks (AONs) 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 them.

There are numerous fault detection mechanisms for traditional electrical communication networks. Unfortunately, such mechanisms cannot be applied directly to 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 transplanted to AONs. Reference [1], for example, has shown that some typical schemes for SDH/SONET could not be applied to AONs.

Therefore new methodologies and mechanisms are

necessary for fault detection in AONs. The optical power detection, optical spectral analysis, pilot tones and optical time domain reflectometry (OTDR) could be deployed for fault detection and also 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 schemes are channel-based and introduces large numbers of monitors thus they are not feasible in today's AONs due to the reasons of channel dynamics, scalability and costs. Other methods like 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 functionality of fault detection. Some key parameters of OSPF 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. Unfortunately the typical fault detection time of routing protocols is at seconds-level, 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 optical layer to IP layer. Thus some effective and efficient fault detection mechanisms at optical layer are still expected.

In this paper, we propose a fault detection mechanism for AONs which utilizing independent wavelengths as supervisory channels. The monitors are assigned based on cycle covers of the network topology. Two cycle cover finding algorithms are developed and compared in terms of the wavelength utilization and the cycle cover cardinality, under four typical example networks: NSFNET, ARPA2, SmallNet and Bellcore.

This paper is organized into the following sections. Section 2 briefly describes the background of cycle cover for bridgeless graphs and introduces the concept of monitoring cycles. Section 3 develops two cycle finding algorithms. Some metrics are defined in Section 4 for

evaluating the algorithms. In Section 5 the algorithms are applied to four network examples and the results are compared in terms of the metrics described in Section 4. Finally some conclusions are drawn in Section 6.

2. Monitoring cycles

The impact scope of failure is various for different network faults: some faults only affect a single or some specific wavelengths, e.g. optical crossconnect port blocking, while others may affect all the wavelengths pass through the faulty module, e.g. fibre cuts. In this paper, we propose a fault detection mechanism for combating the latter. The main idea is to decompose a network into a set of cycles. All nodes and links of the given network are covered at least by one cycle. We define these cycles as “*monitoring cycles*”. A transceiver is assigned to one node in each monitoring cycle and thus a loopback dedicated supervisory channel is set up. Consequently, all of the optical network performance indices, including optical power, optical spectrum, optical signal-to-noise ratio, and more importantly, bit error ratio, can be measured within each monitoring cycle. Once a fault occurs in any link or node belongs to such a cycle, it will trigger an alarm in this monitoring cycle. Such a mechanism can achieve fault detection and isolation in AONs.

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 that 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. Fig. 1 gives an example of a network with 10 nodes and an instance of its cycle covers. 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)*.

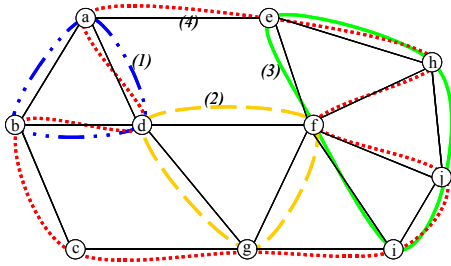


Fig. 1. A graph example and a cycle cover instance

To detect network fault for all nodes and links of a given network, a cycle cover has to be found. At the same time, for maximizing the wavelength utilization, we have also to minimize the number of wavelengths occupied by monitoring channels in all 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 [8],

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

Although the conjecture has not been completely proven, it was shown in [8] that a minimum counterexample to the cycle 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. [9], had conjectured that such snarks did not exist. Thus it is safely 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 regarding the network resource utilization. That is, an achievable minimum average number of occupied wavelengths for monitoring each link is two.

3. Cycle finding algorithms

In this section 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.

3.1. 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 of G 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 $p(y, \dots, x) \in G'$. Thus path $p(y, \dots, x)$ and link $e(x, y)$ consist 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 available, select the uncovered link whose endpoints are also uncovered. Start DFS from e and go to that uncovered endpoint of e if possible;

- 3) At each step of each DFS, select an uncovered link. If multiple links available, alternatively apply the largest/smallest node number first rule, e.g. if in last DFS we select a link whose end-node has the largest number among multiple nodes with same priority, then in the current DFS we select the link whose end-node has the smallest number;
- 4) Once a link returns to the previously visited part, a cycle c can be formed and added 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 selection 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 results of HDFS applying 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 a DFS iteration 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, link $(5,6)$ 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 the remainder iterations. After iteration 5 all links in the graph are covered and a 5-cycle cover is obtained as depicted in Fig. 2.

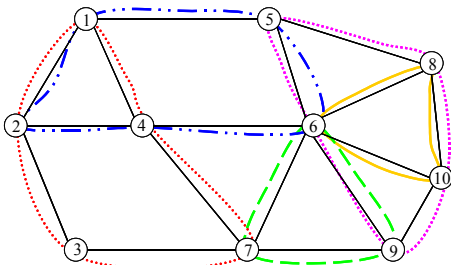


Fig. 2. The cycle cover obtained by HDFS

3.2. 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 sub-cycle. Then we remove this part from the Eulerian cycle and traverse the remainder part until all links are removed. In this way, the Eulerian cycle is decomposed into a cycle cover C consisting of all sub-cycles obtained in the traversing. 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 each added new link corresponds to a path consisting of existing links between the node pair in the original graph. Links included in one augment will be covered one more time in a cycle cover. To minimize the average number of wavelengths occupied by monitoring in links, i.e. minimize the average link cover times, the shortest path augments 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) From V' , start from a node x and find the shortest path to every other node, select the smallest one among them, denote as $p(x, y)$. Add the 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 above-mentioned.

For example, in the graph given in Fig. 1 we firstly label the degree for all nodes and the odd-degree node set is $\{1,2,5,8,9,10\}$. The degrees of all nodes are labelled in Fig. 3(a). For node 1, the shortest path to another node is $(1,5)$ and $(1,2)$, which are a single hop. Select $(1,2)$ and remove node $1,2$ 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(a). 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 in Fig. 3(a). Now all node degrees are even and the augmented graph is Eulerian.

An Eulerian cycle can be found by any existing traversing algorithm, such as DFS. In this example an Eulerian cycle is listed in Fig. 3(b). This Eulerian cycle is decomposed into 4 cycles as shown in Fig. 3(c). Note that a two-edge cycle, e.g. 10-9-10, is not considered as a “real” cycle.

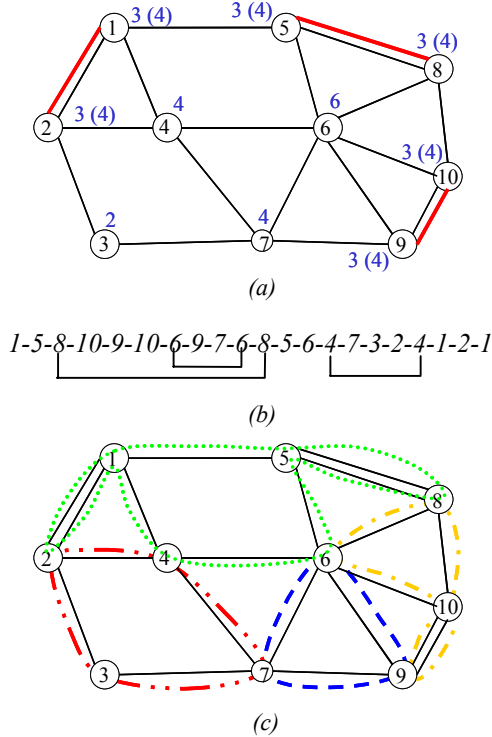


Fig. 3 The graph example for SPEM: (a) The shortest path Eulerian matching; (b) An Eulerian cycle and decomposing; (c) The cycle cover obtained by SPEM

4. Evaluation Metrics

4.1. Effects on wavelength utilization

Since some wavelengths are assigned to monitoring channels within each node and link, the wavelength utilisation of the network will decrease. The maximum number of occupied wavelengths in a node or link is the maximum number of wavelengths assigned for monitoring in this single node or link. It is equivalent to the maximum cover times of a single node/link in a cycle cover. The 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 number of appearance of all nodes or links in a cycle cover.

To quantitatively analyse the relative degradation of wavelength utilization, we defined the maximum and average *wavelength overhead* brought to the network by monitoring cycles per node and per link respectively,

$$H_{\max} = \frac{\text{maximum \# of occupied } \lambda s}{\text{number of total available } \lambda s} \quad (\text{per node or link})$$

$$H_{\text{ave}} = \frac{\text{average \# of occupied } \lambda s}{\text{number of total available } \lambda s} \quad (\text{per node or link})$$

Here H_{ave} indicates the average degradation of wavelength utilization and H_{\max} is for the worst case of wavelength utilization degradation among all nodes/links.

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 [9]. Therefore we set the number of total available wavelengths in a node or link to be 432 for calculating H_{ave} and H_{\max} in example networks given in Section 5.

4.2. Cost analysis

The cardinality is the measurement of the size of a countable set. If a countable set is finite, its cardinality is the number of its elements. A countable but infinite set is said to have the cardinality \aleph_0 . For a finite undirected graph, each cycle cover is a finite set of cycles. The *cardinality of a cycle cover* is defined as the number of cycles in the cover.

In monitoring-cycle based fault detection, since a transceiver is assigned to each cycle and a dedicated supervisory channel is established along the cycle for monitoring (one wavelength is used), the cardinality (cycle number) of a cycle cover represents the total required transceivers as well as the wavelengths for monitoring. Thus it is the measurement of the costs for the fault detection based on monitoring cycles.

5. Examples

We tested and compared the proposed fault detection mechanism using the described two heuristic algorithms respectively upon four example networks: NSFNET, ARPA2, SmallNet, and Bellcore, as shown in Fig. 4. The performance of the two cycle finding algorithms are compared in terms of the metrics described in Section IV. The comparison results are listed in Table 1.

The results in Table 1 have shown that the number of occupied wavelengths per link or node in all examples are small: in maximum 5 λs /node and 3 λs /link using HDFS,

and 3 λ s/node and 2 λ s/link using SPEM. With the assumption of available wavelengths described in Section 4 (432 wavelengths available per link/node), the maximum overhead (in worst case) 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.

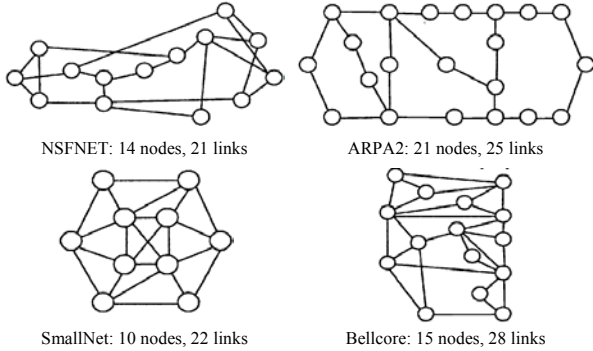


Fig. 4 Topologies of four example networks: NSFNET, ARPA2, SmallNet, and Bellcore

The required number of transceivers for monitoring is determined by the cycle cover cardinality of each network example. We define the ratio between the number of required transceivers and the number of links in a network, to measure the relative costs for fault detection,

$$R_{link} = \frac{\# \text{ of required transceivers for monitoring}}{\# \text{ of links in the network}}$$

The ratios are pretty small for all network examples. For example, the worst case is the SmallNet that R_{link} is

36.4% and 40.9% for using HDFS and SPEM respectively. Comparing to the conventional fault detection schemes, e.g. one-monitor-per-link case, the proposed mechanism cut more than half of the costs of the transceivers. Therefore the proposed fault detection mechanism is cost efficient for meshed AONs.

6. Conclusions and future work

This paper introduced the concept of monitoring cycle and proposed a fault detection mechanism based on decomposing meshed all-optical networks into cycles. Two algorithms, heuristic depth first searching (HDFS) and shortest path Eulerian matching (SPEM), are developed for finding monitoring cycles in AONs. The two algorithms are compared in terms of the node and link wavelength utilization, as well as the number of required transceivers. The results for example networks show that the degradation of wavelength utilization is pretty low and the costs for monitoring are much smaller than one-monitor-per-link scheme. Therefore the proposed mechanism based on monitoring cycles is a promising fault detection method for AONs.

The future work might include the development of more effective 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 potential advantages for the mechanism.

Table 1. Comparison of cycle finding algorithms: HDFS and SPEM

		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	
Cardinality of cycle cover		6	4	4	4	8	9	6	5
R_{link}		28.6%	19.0%	16.0%	16.0%	36.4%	40.9%	21.4%	17.9%
node	Max # of occupied λ s	4	2	3	2	5	3	4	3
	Hmax (node)	0.93%	0.46%	0.69%	0.46%	1.16%	0.69%	0.93%	0.69%
	Ave # of occupied λ s	2.36	1.71	1.62	1.29	3.40	2.50	2.67	1.87
	Have (node)	0.55%	0.40%	0.38%	0.30%	0.79%	0.58%	0.62%	0.43%
link	Max # of occupied λ s	3	2	3	2	3	2	3	2
	Hmax (link)	0.69%	0.46%	0.69%	0.46%	0.69%	0.46%	0.69%	0.46%
	Ave # of occupied λ s	1.57	1.24	1.36	1.20	1.55	1.18	1.43	1.14
	Have (link)	0.36%	0.29%	0.31%	0.28%	0.36%	0.27%	0.33%	0.26%

7. REFERENCES

- [1] Y. Kobayashi, Y. Tada, S. Matsuoka, N. Hirayama, and K. Hagimoto, "Supervisory systems for all-optical network transmission systems," *IEEE Globecom'96*, 1996, pp. 933-937
- [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, 8-12 Nov. 1998, pp. 2309-2314
- [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*, Aug. 2002, pp. 217-223
- [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, Oct. 1997, pp. 1784-1793
- [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] 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
- [9] 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
- [10] S. V. Kartalopoulos, *Fault Detectability in DWDM – Toward Higher Signal Quality & System Reliability*, Piscataway: IEEE Press, 2001