Photon Netw Commun (2006) 11:277–286
DOI 10.1007/s11107-005-7355-3

ORIGINAL ARTICLE

A novel fault detection and localization scheme for mesh all-optical networks based on monitoring-cycles

Hongqing Zeng · Changcheng Huang · Alex Vukovic

Received: 25 February 2005 / Revised: 29 August 2005 / Accepted: 2 September 2005

Abstract We previously showed the feasibility of a fault detection scheme for all-optical networks (AONs) based on their decomposition into monitoring-cycles (m-cycles). In this paper, an m-cycle construction for fault detection is formulated as a cycle cover problem with certain constraints. A heuristic spanning-tree based cycle construction algorithm is proposed and applied to four typical networks: NSFNET, ARPA2, SmallNet, and Bellcore. Three metrics: grade of fault localization, wavelength overhead, and the number of cycles in a cover are introduced to evaluate the performance of the algorithm. The results show that it achieves nearly optimal performance.

Keywords Fault detection · Fault localization · All-optical network · Monitoring cycle · Cycle cover

Introduction

Fault detection and localization are essential for providing continuous and reliable services in all-optical networks (AONs) with ever-increasing data rate as well as increased wavelength number and density in wavelength-division multiplexing (WDM) [1]. For AONs, fault detection and localization can be performed in either the physical or the IP layer. Most routing protocols in the IP layer, e.g., OSPF or IS–IS, have inherently such a functionality [2]. Unfortunately, the long detection time in the IP layer (typical at seconds-level) makes it difficult to achieve time-critical recovery. Thus, some effective and efficient fault detection mechanisms at the optical layer are required. However, existing fault detection and localization mechanisms for conventional networks cannot be applied to AONs directly due to the lack of electrical terminations [3]. Even some detection methods deployed in optical networks with opto-electro-opto (OEO) conversion cannot be transplanted to AONs, for instance the examples in [4]. In the physical layer, network faults can be detected by measuring the optical power, analyzing the optical spectrum, using pilot tones, or performing optical time-domain reflectometry [5]. A fault detection scheme was developed by assigning monitors to the sinks of each optical multiplex section and optical transmission section [6]. Another scheme proposed in [7] modeled all possible states of a link as a finite state machine (FSM). The FSM for each link keeps tracks of the current state of the link by assigning a monitor to the link. Ideally, all potential faults could be completely detected and located by assigning a monitor to each link (channel). However, it is usually not feasible to implement the monitor-per-link scheme in large-scale networks because of the large number of required monitors and the real-time processing of a huge amount of redundant alarms.

Other than assigning a monitor per link, some authors placed a monitor to each established lightpath [8]. Some heuristics were proposed to reduce the number of required monitors based on the information of redundant alarms. This scheme was effective at the time it was proposed, since the

H. Zeng (✉) · C. Huang
Department of Systems and Computer Engineering,
Carleton University,
1125 Colonel By Drive,
Ottawa, ON,
Canada K1S 5B6
e-mail: hzeng@sce.carleton.ca

A. Vukovic
Communications Research Centre Canada,
3701 Carling Avenue,
P.O. Box 11490,
Stn. H, Ottawa, ON,
Canada K2H 8S2
e-mail: alex.vukovic@crc.ca

number of lightpaths in an AON was relatively small and they did not change frequently once established. However, the number of lightpaths soars so much nowadays with the use of DWDM technology that this scheme will introduce huge cost due to the large number of required monitors. Furthermore, most AONs currently support dynamically lightpath provisioning so that the monitor placement has to be dynamically re-calculated and re-located once some lightpaths are changed, which is not easy to fulfill in the real world.

To detect and locate network faults, it is not necessary to put monitors on all links, lightpaths, or nodes. For example, some authors proposed a diagnosis method with sparse monitoring nodes (multiple monitors may be required) particularly for crosstalk attacks, which could be considered as special cases of network faults in a wide sense [9, 10]. In this paper, we propose a general approach at the physical layer for fault detection and localization in AONs through decomposing the given network into a set of cycles, which form a cycle cover for the network. A spanning-tree based cycle construction algorithm is developed and applied to four typical example networks: NSFNET, ARPA2, SmallNet, and Bellcore. The performance of the proposed approach is evaluated in terms of grade of fault localization, costs, and impacts on wavelength utilizations.

This paper is organized into the following sections. Section Monitoring-cycles and cycle construction formulation introduces the concept of monitoring cycles and formulates the problem of constructing monitoring cycles to the cycle cover problem. Section Heuristic spanning-tree (HST) based cycle construction proposes a heuristic spanning-tree based cycle construction algorithm. The proposed algorithm is then applied to four typical example networks in Section Examples and evaluations. The performance measures of the proposed algorithm are also evaluated. Finally, some conclusions are outlined in the last section.

Monitoring-cycles and cycle construction formulation

We previously proposed a fault detection and performance-monitoring scheme based on decomposing an AON into a set of cycles [11]. All nodes and links in the network appear in at least one of these cycles, which form a cycle cover of the network. A network monitor is assigned to one node in each cycle and a loopback supervisory channel is set up in this cycle. A cycle with a monitor and a supervisory channel is defined as “monitoring cycle (m-cycle)”. Network faults trigger alarms in m-cycles that cover the faulty source thus they are detectable. Depending on the type of monitors in the m-cycles (e.g., optical power meters, optical spectrum analyzers, and transceivers), various performance parameters of AONs can be measured, such as optical power, channel wavelength, optical signal-to-noise ratio, and bit error ratio. Flexible index thresholds can be set to determine whether a network fault occurs. Such an approach reduces the number of required monitors from the number of links to the number of cycles in an AON. Furthermore, the loopback supervisory scheme puts the transmitter and receiver together in a single node of an m-cycle. Thus the source signals could be used as references for received signals and the hardware and software of monitoring devices could be greatly simplified. More importantly, through assigning monitors to some selected nodes and/or avoiding assigning monitors to the nodes with high management expense, the cost of our fault detection and localization mechanism could be kept in low.

A meshed AON can be modeled as a finite undirected graph $G(V, E)$, where V is the set of vertices (nodes) and E is the set of edges (links). Hereafter the term vertex (edge) and node (link) are exchangeable in this paper. We assume that such a graph is connected and it contains neither loops nor multiple edges. A loop is an edge that starts and ends at the same vertex. Multiple edges refer to two or more parallel edges that have the same start-vertex and end-vertex. Furthermore, an edge is a bridge of a graph if the graph becomes from a connected graph to be a disconnected one after deleting it. A bridge link is a single-failure point for the network, thus it is usually avoided during the network topology design. Therefore, $G(V, E)$ is assumed to be bridgeless.

A cycle (denoted as c) of the graph G is a sub-graph of G that is connected and regular of degree two. It is often identified with its edge-set. A cycle cover (denoted as C) of a graph is a set of cycles in which each vertex and edge of the graph appears at least in one of these cycles. According to the m-cycle definition, the set of m-cycles is a cycle cover for a given graph. Let $C = \{c_1, c_2, \ldots, c_M\}$ be such a set of m-cycles. For an edge $e \in E$, let $C(e)$ denote the number of cycles in C that contain e, that means $C(e) = |\{i: e \in c_i\}|$ where $|\bullet|$ represents the set cardinality, i.e., the number of elements in a finite set. When $C(e) = t$, we say that the cover time of edge e is t in C. The length of a cycle is the number of edges it contains, denoted by $\text{len}(c_i) = |c_i|$. The length of C, denoted as $\text{len}(C)$, is the sum of all cycle lengths in C. Obviously we have,

$$\text{len}(C) = \sum_{i=1}^{M} |c_i| = \sum_{j=1}^{L} C(e_j). \tag{1}$$

While looking for a set of m-cycles (a cycle cover) for a graph, we have to take the following considerations into account: grade of fault localization, wavelength overhead due to m-cycles, and cost of the required monitors.

Grade of fault localization

A network fault triggers alarms in the m-cycles in which it appears, but not others. Reversely, if alarms are received in some

m-cycles but no others, it implies that the potential faulty links are the common links of these m-cycles. Figure 1 gives a graph example with a cycle cover $C = \{c_1, c_2, c_3, c_4\}$. If, for example, a fault occurs along link (1,4), it will trigger alarms in c_1 and c_4, respectively, but there is no alarm in other m-cycles. If alarms are received in m-cycles c_1 and c_4, but not in others, it implies that the potential faulty links could be either (1,4) or (2,4) or both, since the alarm distribution triggered by the fault in both links are identical, which means alarms in c_1 and c_4. Generally, a binary indication bit m_j can be defined for m-cycle c_j to indicate whether or not a fault occurs and thus an alarm appears in it,

$$m_j = \begin{cases} 1 & \text{an alarm appears in } c_j \\ 0 & \text{no alarm appears in } c_j \end{cases} ; \quad j = 1, 2, \ldots, M. \quad (2)$$

The sequence of such bits for a link forms an alarm code (M bits in total). Alarms are sent to a centralized network management unit (NMU) and alarm codes are generated in real time. Furthermore, for any link $e_i \in E (i = 1, 2, \ldots, L)$ and m-cycle c_j, a binary associative bit a_{ij} is defined as,

$$a_{ij} = \begin{cases} 1 & e_i \text{ is covered by } c_j \\ 0 & e_i \text{ is not covered by } c_j, \end{cases} \quad (3)$$

where $i = 1, 2, \ldots, L$ and $j = 1, 2, \ldots, M$. The sequence of associative bits of a link corresponding to all the m-cycles forms the associative code (M bits in total). Once alarms in m-cycles are collected and an alarm code is generated, we can compare the alarm code bit-by-bit with the associative code for each link. If a link's associative code exactly matches the alarm code, then this link is a faulty candidate for the received alarm code. By matching the real-time alarm codes with the associative codes of all links, a faulty candidate set can be established for each alarm code. Based on the selection of m-cycles, multiple elements may exist in such candidate sets. To quantitatively measure the grade of localization, we introduce the concept of *Localization Degree* (denote as I), which is defined as the average size of non-empty faulty candidate sets produced by all possible alarm codes. Let $C = \{c_1, c_2, \ldots, c_M\}$ be a set of m-cycles in graph G. Since each alarm code consists of M bits, the number of possible alarm codes is $2^M - 1$. Let s_k be the faulty candidate set for alarm code $\mathbf{m}_k$, where $k = 1, 2, \ldots, 2^M - 1$. Please note for a given AON, some alarm codes are not applicable and thus the corresponding faulty candidate sets are empty. Let D be the collection of all non-empty s_k. Then the localization degree can be defined as the following,

$$I = \frac{\sum_{s_k \in D} |s_k|}{|D|}. \quad (4)$$

In the ideal case, every candidate set has only one element and $I_{\text{ideal}} = 1$ (defined as complete localization). In building m-cycles for fault detection, we want to minimize the localization degree, i.e., MIN I.

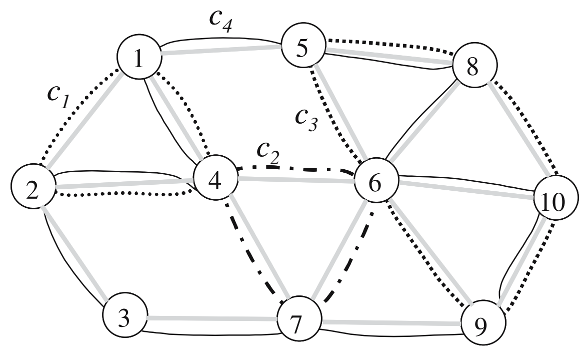

Fig. 1 A graph example and cycle cover

Wavelength overhead

In each link, some wavelength channels are reserved for m-cycles. These channels cannot be used for carrying user traffic and therefore become an overhead. The number of reserved wavelengths within a link is equal to the cover times of that link in the cycle cover. Let $\Lambda_{\max}$ be the maximum number of wavelengths reserved for monitoring in a link and it represents the worst case. Given a graph with N vertices, L edges and M m-cycles, the average number of reserved wavelengths (Λ_{avg}) for all edges is equal to the average cover time and,

$$\Lambda_{\text{avg}} = \sum_{i=1}^{L} C(e_i) \big/ L = \text{len(C)} \big/ L. \quad (5)$$

To quantitatively analyze the relative overhead due to m-cycles, we define the average wavelength overhead per link brought to the network by m-cycles as $\text{WOH}_{\text{avg}} = \Lambda_{\text{avg}}/F$, where F is the number of total available wavelengths per link. To minimize the wavelength overhead, we have to minimize Λ_{avg}, which is equivalently to minimize the cycle cover length. Consequently, m-cycle construction can also be formulated to the least cost cycle cover problem for un-weighted graphs.

Cost of monitors

Since a monitor and a dedicated supervisory channel are assigned for each m-cycle, the number of required monitors and reserved wavelengths, i.e., the number of m-cycles ($M = |C|$), is a measure of the cost for such type of fault detection and localization approaches. To minimize this cost, we have to minimize the number of m-cycles, i.e., $MIN \ |C|$.

Heuristic spanning-tree (HST) based cycle construction

It has been proven that cycle covers exist for each bridgeless, connected, undirected graph and could be obtained in running time $O(N^2)$ [12]. Numerous algorithms have been reported for constructing cycle covers, e.g., a polynomial-time algorithm was proposed in [13]. Unfortunately, such works are focused on the least-cost cycle cover problem but do not consider the localization degree and cycle numbers. We previously developed two m-cycle construction algorithms: heuristic depth-first searching and shortest path Eulerian matching algorithm, with a balance of all three considerations [11]. In this paper, we propose a heuristic spanning-tree based m-cycle construction algorithm for the same purpose, while improving the performance in terms of localization degree.

Preliminary

For a connected, bridgeless, simple graph $G(V, E)$, there must exist a spanning-tree T. For each link $e \notin T$ (whereby e is called a "chord"), it holds that if the two endpoints of e are n_1 and n_2, then $n_1, n_2 \in T$ and there must exist a path $p \in T$ connecting n_1, n_2. Thus, link e and path p form a cycle. We say this cycle is generated by chord e. Each chord generates such a unique cycle. We also have the following cycle cover existence lemma [14].

Lemma There exists at least one cycle cover for a bridgeless graph $G(V, E)$.

For a cycle c_i in a graph G with precisely L edges, an associative vector $\mathbf{v_i}$ with L components can be assigned to it. The jth component v_i^j of the vector $\mathbf{v_i} = (v_i^1,\ v_i^2, \ldots, v_i^L)$ is *one* if the jth edge of G lies in c_i, and *zero* otherwise (please note the difference with the associative code defined in Section Grade of fault localization). Cycles $c_1,\ c_2, \ldots, c_M$ are called independent if their associative vectors are linearly independent, where a group of vectors $\mathbf{a_1},\ \mathbf{a_2}, \ldots, \mathbf{a_n}$ are linearly independent if and only if $\sum_{i=1}^{i=n} k_i \mathbf{a_i} = 0$ holds when $k_1 = k_2 = \cdots = k_n = 0$.

H. Walther has proven the following theorem in [13],

Walther's Theorem: Let G be a connected graph with L edges and N vertices. Then there exist $L - N + 1$, but no more independent elementary cycles.

A cycle is called elementary cycle if no vertex is encountered more than once when traversing it. Each cycle can be partitioned into elementary cycles. In this paper, a cycle refers to an elementary cycle

Based on the above lemma and theorem, we claim the following theorem,

New theorem: For a connected, bridgeless, simple graph G with a given spanning-tree, cycles generated by all chords construct a cycle cover for G.

Proof Let G has L edges, N vertices, and the spanning-tree is T. Then, all edges can be partitioned into two sets: $N - 1$ edges in T and $L - N + 1$ edges not in T.

(1) Each edge not in T is a chord. It generates a cycle and is covered by this cycle. There are $L - N + 1$ such cycles.

(2) Assume that there exists an edge $e^* \in T$ and e^* is not covered by any cycles generated by those chords. Because of the lemma, there must exist another cycle c_0 in which e^* appears. The associative vector of c_0 is $\mathbf{a_0} = (a_0^1,\ a_0^2, \ldots, a_0^*, \ldots, a_0^L)$, where $a_0^* = 1$ and corresponding to the position of edge e^*. For all other cycles, the components at this position of the associative vectors are zero, because they do not cover edge e^*. Thus, cycle c_0 is independent from all other $L - N + 1$ cycles. Consequently, by adding c_0, graph G now has $L - N + 2$ independent cycles. But Walther's theorem indicates that there are no more than $L - N + 1$ independent cycles in graph G. □

In the proof, please note that a cycle cover of graph G is uniquely determined by the given spanning-tree.

Heuristic spanning-tree (HST) based algorithm

Among the numerous existing cycle building algorithms, the spanning-tree based ones are fast, simple, and flexible. Breadth-first and depth-first spanning-trees (BFST and DFST) are well known and have been in common use for a long time. Numerous algorithms to generate such spanning-trees have been intensively studied [15]. Figure 2 gives three spanning-trees for an example graph and m-cycles generated by corresponding chords. Numbers of cycles in the covers generated by various spanning-trees are the same for the graph. By enumerating the faulty candidates for all possible alarm codes, we find that their localization degrees are also the same ($I = 1$). However, the figure shows that the average cover time (i.e., the average wavelength overhead) per link is smaller for the cover generated by BFST than by DFST. Furthermore, the average cover time might be decreased by including nodes with large degrees in the tree (comparing Fig 2 b and c). This observation leads us to choose BFST and apply a heuristic rule of putting the large-degree nodes into the spanning-tree as early as possible while generating the spanning-tree for constructing a cycle cover.

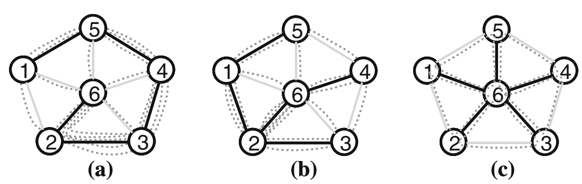

Fig. 2 Spanning-tree and cycle cover. (**a**) DFST: max cover time = 5/link, average cover time = 2.3/link. (**b**) BFST (rooted from random nodes): max cover time = 5/link, average cover time = 1.9/link. (**c**) BFST (rooted from the node with maximum degree): max cover time = 2/link, average cover time = 1.5/link

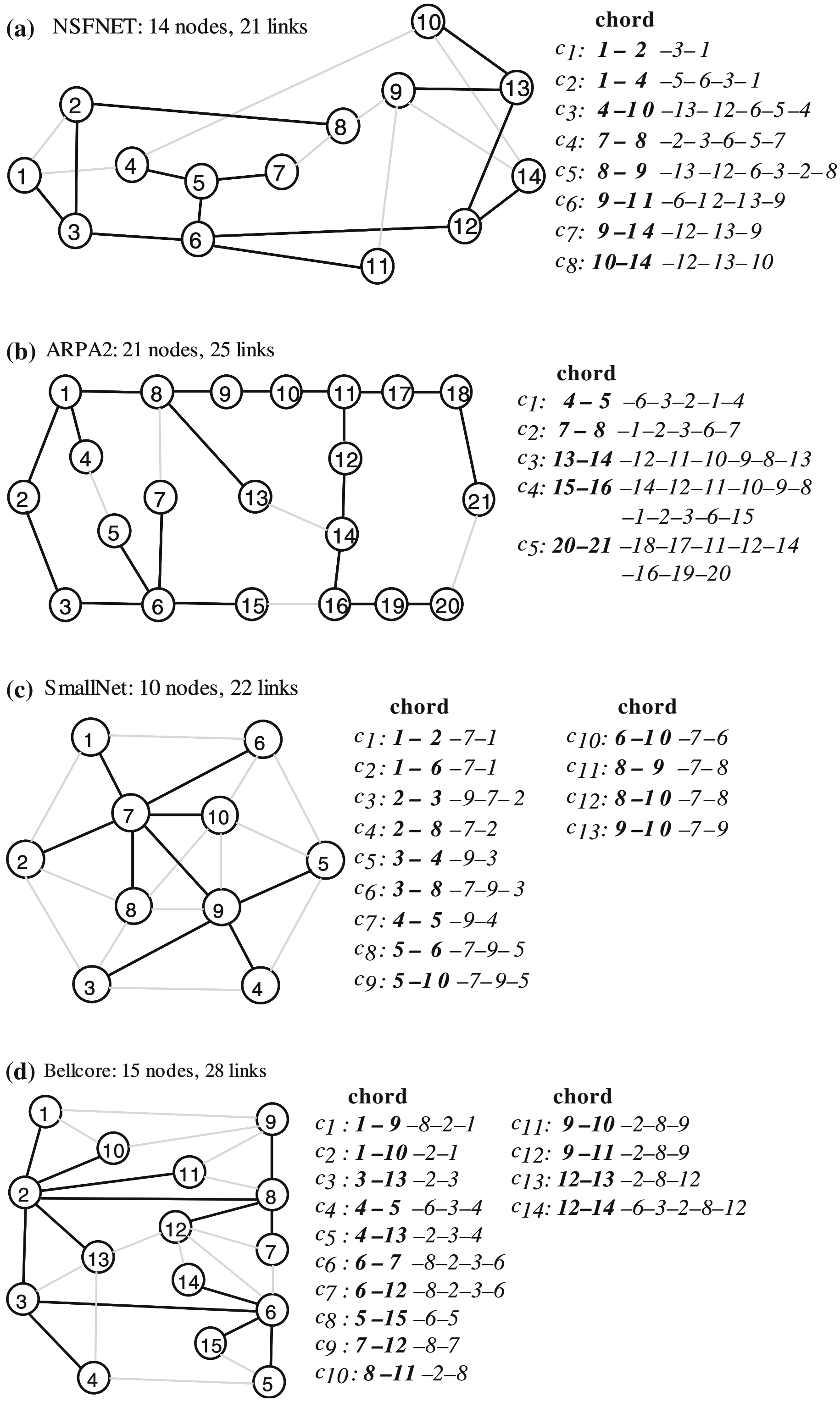

Fig. 3 *m*-Cycles obtained by HST for (**a**) NSFNET; (**b**) ARPA2; (**c**) SmallNet; (**d**) Bellcore (links in spanning-trees are in dark)

A heuristic spanning-tree (HST) based cycle construction algorithm is then given below,

1. Initial: for a graph G, label the degrees of all nodes; set the spanning-tree $T = null$; select the node with the maximum degree as the root. Add all links to T that are incident to the root.
2. For each node $n_i \in T$, update its degree label with the number of links that are incident to n_i and connect n_i with nodes not in T.
3. Select the node with the maximum degree label in T. Add all links to T that are incident to the selected node and connect it with nodes not in T.
4. Repeat steps 2–3 until all node-degree labels are zero. Now T is a spanning-tree of G.
5. Given T, construct the cycles for all chords. They form a cycle cover and are the required m-cycles.

For fault localization based on m-cycles obtained by the HST algorithm, each chord appears in a unique m-cycle and thus it can be completely localized if faults occur upon such edges. Most edges in T are also completely localizable for topologies of real telecommunication networks, although it is not guaranteed all the time. Therefore, the lower bound ratio of localizable links is

$$\text{(localizable link)\%} = \frac{L - N + 1}{L} = 1 + 1\big/L - 2\big/\bar{d}, \quad (6)$$

where $\overline{d} = 2N\big/L$ is the average node degree of graph G. This lower bound shows that the ratio of localizable links in a graph is in inverse proportion to the average node degree. It implies that the HST algorithm has better performance in terms of localization degree for more complex networks.

Examples and evaluations

In this section, the HST based cycle construction algorithm is applied to four typical example networks (NSFNET, ARPA2, SmallNet, and Bellcore). The network topologies, spanning-trees, and m-cycles obtained by the HST algorithm are shown in Fig. 3. The performance of the algorithm is evaluated in terms of localization degree, cost, and wavelength overhead.

Tables 1–4 enumerate all possible alarm codes and corresponding faulty candidate sets for the four example networks, respectively. Table 5 summarizes these localization results and compares them with the heuristic depth-first searching (HDFS) and the shortest-path Eulerian matching (SPEM) algorithms reported in [11]. The comparison shows that the HST algorithm has much better performance than the HDFS and SPEM algorithms in terms of fault localization degree. Further analyses indicate that the HST algorithm performs even better in terms of localization degree for graphs with larger average node degrees. More specifically, for graphs with average node degree larger than 3.0, the localization degrees of the HST algorithm are very close to the ideal case ($I_{\text{ideal}} = 1$). This observation implies that such fault detection and localization approaches are suitable for complex networks (with large average node degree), and thus are scalable.

Table 1 Fault localization results: NSFNET — HST

Alarm code								Fault candidate
c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	
0	0	0	0	0	0	0	0	Null
							1	10–14
						1		9–14
						1	1	12–14
					1			6–11, 9–11
				1				8–9
				1	1	1		9–13
			1					5–7, 7–8
			1	1				2–8
		1						4–10
		1					1	10–13
		1		1	1			6–12
		1		1	1	1	1	12–13
	1							1–4
	1		1	1				3–6
	1	1						4–5
	1	1	1					5–6
1								1–2
1			1	1				2–3
1	1							1–3
			Others					N/A

Table 2 Fault localization results: ARPA2 — HST

Alarm code					Fault candidate
c_1	c_2	c_3	c_4	c_5	
0	0	0	0	0	Null
				1	11–17, 16–19, 17–18, 18–21, 19–20, 20–21
			1		6–15, 15–16
			1	1	14–16
		1			8–13, 13–14
		1	1		8–9, 9–10, 10–11
		1	1	1	11–12, 12–14
	1				6–7, 7–8
	1		1		1–8
1					1–4, 4–5, 5–6
1	1		1		1–2, 2–3, 3–6
		Others			N/A

Table 3 Fault localization results: SmallNet — HST

Alarm code													Fault candidate
c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9	c_{10}	c_{11}	c_{12}	c_{13}	
0	0	0	0	0	0	0	0	0	0	0	0	0	Null
												1	9–10
											1		8–10
										1			8–9
									1				6–10
								1					5–10
								1	1		1	1	7–10
							1						5–6
						1							4–5
						1	1	1					5–9
					1								3–8
				1									3–4
				1		1							4–9
			1										2–8
			1		1					1	1		7–8
		1											2–3
		1			1		1	1		1		1	7–9
		1		1	1								3–9
	1												1–6
	1						1		1				6–7
1													1–2
1		1	1										2–7
1	1												1–7
						Others							N/A

The cost of the proposed scheme is measured by the number of required monitors and reserved wavelengths for the m-cycles. The cost of the wavelengths is evaluated by Λ_{avg}, Λ_{max}, and WOH_{avg}, as described in Section Monitoring-cycles and cycle construction formulation. In Table 6, the maximum and average numbers of reserved wavelengths in the network links are summarized and compared with the HDFS/SPEM algorithms. It shows that the numbers of both maximum and average reserved wavelengths for the m-cycles obtained by HST algorithm are larger than when performing HDFS and SPEM. This is the payment for the benefit in localization degree. Nevertheless, with DWDM technology, the number of wavelengths in a single link tends to become larger. For example, it was reported already in 2001 that 432 wavelengths could be multiplexed into a single fiber [16, 17]. In current commercial DWDM systems, it is easy to boost the number of available wavelengths in a fibre to 192 or above [18]. Even for a small number of available wavelengths per link, e.g., $F = 64$, the wavelength overhead for the HST algorithm is small (around 3%, see Table 6). Such overhead has trivial impact on network utilization, if it is not negligible.

The cost of monitors is weighted by the number of monitors for the m-cycles, i.e., the number of m-cycles (denoted as M). For comparing with the monitor-per-link case, a cost gain is calculated as $G = (L - M)/L$, where L is the number of links. The cost gains of the HST algorithm are

Table 4 Fault localization results: Bellcore — HST

Alarm code														Fault candidate
c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9	c_{10}	c_{11}	c_{12}	c_{13}	c_{14}	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	Null
													1	6–14, 12–14
												1		12–13
											1			9–11
										1				9–10
									1					8–11
									1		1			2–11
								1						7–12
							1							5–15, 6–15
						1								6–12
						1		1				1	1	8–12
					1									6–7
					1			1						7–8
				1										4–13
			1											4–5
			1				1							5–6
			1		1	1							1	3–6
			1	1										3–4
		1												3–13
		1		1								1		2–13
		1		1	1	1							1	2–3
	1													1–10
	1									1				2–10
1														1–9
1										1	1			8–9
1					1	1			1	1	1	1	1	2–8
1	1													1–2
Others														N/A

Table 5 Comparison of localization degree

Network example	Avg. node degree	Algorithm	Localization degree	Max candidate set size
NSFNET	3.00	HST	1.105	2
		HDFS	1.50	3
		SPEM	3.00	7
ARPA2	2.38	HST	2.500	6
		HDFS	3.13	6
		SPEM	5.00	8
SmallNet	4.40	HST	1.000	1
		HDFS	1.47	3
		SPEM	3.67	6
Bellcore	3.73	HST	1.077	2
		HDFS	2.15	6
		SPEM	4.67	8

Table 6 Comparison of wavelength overhead

Λ_{max} : the maximum number of wavelengths reserved for monitoring in a link Λ_{avg} : the average number of reserved wavelengths for all links WOH_{avg} : average wavelength overhead per link. It is calculated as $WOH_{avg} = \Lambda_{avg}/F$, where F is the number of total available wavelengths per link. In this table WOH_{avg} is calculated for $F = 64$

Network example	Algorithm	Λ_{max}	Λ_{avg}	WOH_{avg}(%)
NSFNET	HST	5	1.90	2.97
	HDFS	3	1.57	2.45
	SPEM	2	1.24	1.94
ARPA2	HST	3	1.60	2.50
	HDFS	3	1.36	2.13
	SPEM	2	1.20	1.88
SmallNet	HST	6	1.95	3.05
	HDFS	3	1.55	2.42
	SPEM	2	1.18	1.84
Bellcore	HST	8	1.96	3.06
	HDFS	3	1.43	2.23
	SPEM	2	1.14	1.78

Table 7 Comparison of cost gains

Network example	Algorithm	M	$G(\%)$	M'	$M+M'$	$G'(\%)$
NSFNET	HST	8	61.9	2	10	52.4
	HDFS	6	71.4	7	13	38.1
	SPEM	4	80.9	15	19	9.5
ARPA2	HST	5	80.0	15	20	20.0
	HDFS	4	84.0	16	20	20.0
	SPEM	4	84.0	18	22	12.0
SmallNet	HST	13	40.9	0	13	40.9
	HDFS	8	63.6	7	15	31.8
	SPEM	4	81.8	16	20	9.1
Bellcore	HST	14	50.0	2	16	42.9
	HDFS	6	78.6	15	21	25.0
	SPEM	5	82.1	21	26	7.1

M : the number of m-cycles
M' : the number of extra monitors for achieving complete fault localization
G : cost gain over the monitor-per-link method
G' : revised cost gain over the monitor-per-link method under the complete fault localization

compared in Table 7 with HDFS and SPEM for the example networks. Because the monitor-per-link approach always achieves complete localization, for a fair comparison, we add some extra monitors for those links that cannot be fully localized under the HST algorithm to achieve complete localization. For example, a straightforward method would be the following one. If there are $K \geqslant 2$ links in a faulty candidate set, we assign $K-1$ extra monitors to $K-1$ of those K links. More efficient methods might be applied for achieving complete localization, e.g., using extra m-cycles. Therefore, the HST algorithm still has good cost gains, although the M values of HST are larger than HDFS and SPEM. Denote the number of extra monitors as M', the complete localization can be achieved and the cost gain calculation is revised as,

$$G' = (L - (M + M'))\big/L. \tag{7}$$

Revised cost gains obtained from the four example networks are also compared in Table 7. Again, the average node degree affects the cost gain. For graphs whose average node degree is 3.0 or above, the cost gain for HST is 40–52%. For the worst case, ARPA2, it still achieves a cost gain no less than 20%. Such results show that under a fair comparison, the cost gains of HST are better than those of HDFS and SPEM.

The fault detection scheme in [8], as described in Section Introduction, placed a monitor per path. In a N-node network, typically each node has to communicate with all the others. Thus, the number of potential paths is $N(N-1)$. Even with 50% savings of monitors (in maximum) by applying the proposed heuristic optimization algorithm, the number of required monitors is still $O(N^2)$. Clearly, the m-cycle based approach achieves significant cost gains in all examples compared to either the monitor-per-link or the monitor-per-path case.

Conclusion

In mesh AONs, network faults can be detected and located by decomposing them into monitoring-cycles (m-cycles). We formulated the m-cycle construction as a cycle cover problem with certain constraints. A heuristic spanning-tree (HST) based m-cycle construction algorithm has been developed and evaluated in terms of localization degree, wavelength overhead, and cost gain. The proposed HST algorithm has been applied to four typical networks (NSFNET, ARPA2, SmallNet, and Bellcore) and compared to the previously reported algorithms, HDFS and SPEM. The comparison results show that the performance of localization degree for HST algorithm is better than HDFS and SPEM. Analyses indicate that the average node degree of a network plays an important role in the performance of m-cycle based fault detection and localization approaches. The fact that m-cycle based approaches can achieve better performance in networks with a larger average node degree implies that such approaches are suitable for complex networks and thus scalable.

The HST algorithm introduces more monitors than in HDFS and SPEM. However, in a fair comparison of achieving complete localization, it has better cost gains than HDFS and SPEM. Additionally, all the three m-cycle construction algorithms have good cost gains over either monitor-per-link or monitor-per-path case. Finally, the wavelength overheads due to m-cycles are negligible in all approaches. Therefore, the HST algorithm is effective and cost-efficient.

References

1. Ramamurthy, S., Sahasrabuddhe, L., Mukherjee, B.: Survivable WDM mesh networks, IEEE/OSA J. Lightwave Technol. **21**(4), 870–883 (2003)
2. Goyal, M., Ramakrishnan, K.K., Feng, W.-C.: Achieving faster failure detection in OSPF networks. Proc. of IEEE ICC'03, Anchorage, Alaska, USA, May 2003, vol. 1, pp. 296–300
3. Mas, C., Thiran, P.: A review on fault location methods and their applications in optical networks. Optical Network Magazine **2**(4), 73–87 (2001)
4. Kobayashi, Y., Tada, Y., Matsuoka, S., Hirayama, N., Hagimoto, K.: Supervisory systems for all-optical network transmission systems. Proc. of IEEE Globecom'96, London, UK, November 1996, vol. 2, pp. 933–937

5. Mèdard, M., Marquis, D., Chinn, S.R.: Attack detection methods for all-optical networks. Proc. of Network and Distributed System Security Symposium, San Diego, California, March 1998, session 3
6. Hamazumi, Y., Koga, M., Kawai, K., Ichino, H., Sato, K.: Optical path fault management in layered networks. Proc. of IEEE Globecom'98, Sydney, Australia, November 1998, vol. 4, pp. 2309–2314
7. Li, C.-S., Ramaswami, R.: Automatic fault detection, isolation, and recovery in transparent all-optical networks. IEEE/OSA J. Lightwave Technol. **15**(10), 1784–1793 (1997)
8. Stanic, S., Subramaniam, S., Choi, H., Sahin, G., Choi, H.-A.: On monitoring transparent optical networks. Proc. of International Conference on Parallel Processing Workshops, Vancouver, British Columbia, Canada, August 2002, pp. 217–223
9. Wu, T., Somani, A.K.: Attack monitoring and localization in all-optical networks. Proc. of SPIE Opticommun 2002: Optical Networking and Communications, Boston, MA, USA, July 2002, vol. 4874, pp. 235–248
10. Wu, T., Somani, A.K.: Necessary and sufficient condition for k crosstalk attacks localization in all-optical networks. Proc. of IEEE Globecom'03, San Francisco, CA, USA, Dececember 2003, vol. 5, pp. 2541–2546
11. Zeng, H., Huang, C., Vukovic, A., Savoie, M.: Fault detection and path performance monitoring in meshed all-optical networks," Proc. of IEEE Globecom'04, Dallas, TX, USA, November 2004, vol. 3, pp. 2014–2018
12. Itai, A., Rodeh, M.: Covering a graph by circuits, *Automata, languages and programming,* Lecture Notes in Computer Science vol. 62, pp. 289–299, Springer-Verlag, Berlin, (1978)
13. Fan, G.: Covering graphs by cycles. SIAM J. Discrete Mathe. **5**(4), 491–496 (1992)
14. Walther, H.: Ten applications of graph theory, Chapter 1: Flows and tensions on networks. D. Reidel Pub. Co. Boston, (1984)
15. Wilson, R.J., Watkins, J.J.: Graphs: an introductory approach, Chapter 10: Trees, Wiley, New York (1990)
16. Kartalopoulos, S.V.: Fault detectability in DWDM – toward higher signal quality & system reliability, IEEE Press, Piscataway, (2001)
17. Grover, W.D.: Mesh-based survivable networks: options and strategies for optical, MPLS, SONET, and ATM Networking, Prentice Hall PTR, Upper Saddle River, NJ (2004)
18. Elliott, R.: Dark fibre pricing analysis Europe 1998–2002, [Online document], November 2002, available at http://www.band-x.com/information/Dark_Fibre_Report-prices_98–02webversion.pdf

Hongqing Zeng received the B. Eng. degree in Electrical Engineering from Huazhong University of Science and Technology, Wuhan, P.R. China, in 1990. He received the M.Sc. degree in Electrical Engineering from Wuhan University, Wuhan, P.R. China, in 1995. He worked for Industrial and Commercial Bank of China from 1995 to 2000 as a research engineer. He was a research engineer from 2002 to 2004 in the Broadband Network Technologies Research Branch of Communications Research Center, Ottawa, Canada. Currently he is a Ph.D. candidate in Electrical Engineering, in the Department of Systems and Computer Engineering, Carleton University, Ottawa, Canada. His research interest is optical communication networks.

Dr. Changcheng Huang received B. Eng. in 1985 and M.Eng. in 1988 both in Electronic Engineering from Tsinghua University, Beijing, China. He received a Ph.D. degree in Electrical Engineering from Carleton University, Ottawa, Canada in 1997. He worked for Nortel Networks, Ottawa, Canada from 1996 to 1998 where he was a systems engineering specialist. From 1998 to 2000 he was a systems engineer and network architect in the Optical Networking Group of Tellabs, Illinois, USA. Since July 2000, he has been with the Department of Systems and Computer Engineering at Carleton University, Ottawa, Canada where he is currently an associate professor. Dr. Huang won the CFI new opportunity award for building an optical network laboratory in 2001. He is currently an associate editor of IEEE Communications Letters.

Dr. Alex Vukovic has over 20 years in research and technology development leadership in optical communications, network architecture and thermal management acquired at university, industry, and research laboratories. Alex's contributions involve over 60 journal and conference papers, patents, white papers, technology roadmaps and book chapters. In addition, Alex is an internationally recognized speaker, conference chair, invited university lecturer, member of numerous international research lecturer, member of numerous international research committees, industry program reviewer, scientific committee chairman, keynote speaker and scientific authority. For his achievements, Nortel Networks presented him with a Gold Award.

Currently at the Communications Research Centre (Ottawa, Canada), Alex's focus is on research leadership to verify and validate network concepts and key building blocks for next generation communication networks. He is also an Adjunct Professor at the University of Ottawa. Alex earned his M.Sc. and Ph.D. degrees from the University of Belgrade, Yugoslavia, in 1987 and 1990, respectively.