J Netw Syst Manage (2012) 20:76–96
DOI 10.1007/s10922-011-9206-0

Adaptive Fault Monitoring in All-Optical Networks Utilizing Real-Time Data Traffic

Dong Shen · Kam-Hon Tse · Chun-Kit Chan

Received: 14 September 2010 / Accepted: 7 August 2011 / Published online: 26 August 2011

Abstract We propose a novel fault detection and localization scheme for all-optical networks with the information of real-time data traffic. Our adaptive fault localization framework is based on combining passive and proactive monitoring solutions, together with adaptive management in two phases. Numerical results indicate that our proposed scheme has good scalability, in terms of the number of fault monitors required. Also, we show that our framework allows more flexible network design, and requires much less monitoring bandwidth when compared with the passive monitoring solutions.

Keywords Fault management · Network monitoring · Fault localization · Label monitor

1 Introduction

Owing to the tremendous bandwidth demand of Internet traffic, optical fiber, with its vast transmission capacity, is the only promising transmission medium for backbone networks. With the recent development of the optical fiber transmission technology and the wavelength division multiplexing (WDM) technique, reduction in optical component costs, as well as the transparency to diverse modulation formats and protocols, has enhanced the feasibility and practicality of all-optical networks [1].

D. Shen · K.-H. Tse · C.-K. Chan (✉)
Department of Information Engineering, The Chinese University of Hong Kong,
Shatin, N.T., Hong Kong, China
e-mail: ckchan@ie.cuhk.edu.hk

D. Shen
e-mail: sd009@ie.cuhk.edu.hk

K.-H. Tse
e-mail: tkh008@ie.cuhk.edu.hk

Nevertheless, all-optical networks are vulnerable to physical failures, such as fiber cuts, optical cross-connect (OXC) malfunctions and optical amplifier breakdowns [2]. Due to the extremely large transmission capacity of all-optical networks, these possible failures may be translated as disastrous communication disruptions. Hence, fault management is one of the crucial aspects in network management to assure network reliability and availability. With the increased complexity of the network topology, fault detection and localization may incur significant management and operating costs. Thus, an efficient and cost-effective fault detection and localization system is highly desirable to assure the specified levels of quality of service [3]. In this paper, we propose a novel adaptive fault monitoring framework to fulfill these requirements in all-optical networks.

In all-optical networks, fault localization is more complicated than that in opaque networks as the impact of a single fault may propagate without an electronic boundary. Until recently, there have been two major solutions to monitor link failures in all-optical networks, namely passive detection and localization [4], as well as proactive detection and localization [5]. Passive monitoring solution places equipment called *monitor* to collect fault alarms over the whole network and localize the failure according to the alarms received from all monitors. Link-based monitoring is the most straightforward and conventional passive monitoring solution that requires one monitor per link. To reduce the number of monitors, the concept of a monitoring cycle (m-cycle) [6] has been introduced. However, as the network topology is becoming more complex, much effort has to be made to design a feasible m-cycle coverage solution. Furthermore, even if a feasible cycle set is found, the cost of the required monitors is quite substantial. Besides, the extra monitoring bandwidth cost is also getting high, as each monitoring cycle requires one distinct wavelength in an all-optical WDM network. On the other hand, the authors in [5] have proposed an efficient failure detection and localization scheme by sending out a series of proactive probes with perfect feedback. The idea behind proactive detection is to fully utilize the unique property of an all-optical lightpath. In other words, one lightpath could detect a number of consecutive fiber links simultaneously, if no failure is incurred. With the proactive monitoring scheme, it has been proved that the probing effort can achieve approximately the entropy of the network state under the probability link failure model assumption, in terms of the information theory. Nevertheless, this proactive monitoring scheme may not be practical to be employed in practical transparent optical networks. First, this proactive probing scheme can only be applied to Eulerian networks, each of which contains an Euler trail (a path containing all the links without repetition). Although the authors in [5] have further improved their scheme to accommodate node failures and have demonstrated that all network topologies could be transformed to Eulerian networks, it is still unrealistic to configure the switching nodes to meet the requirements, as it may disrupt the existing connections and may largely increase management costs. Besides, the requirement on the probing time and frequency is still not yet resolved.

Motivated by the pros and cons of both passive and proactive detection solutions, we have developed a novel fault detection and localization framework that utilizes the concept of adaptive network management. Compared with the passive monitoring solutions, our proposed adaptive solution requires fewer monitors and incurs minimal bandwidth costs. Meanwhile, our framework is complete and practical, when compared with the proactive monitoring scheme in [5]. In general, our framework makes use of the real-time network traffic, whose routes are flexible to form the real monitoring trails, to passively monitor the network link states, and then proactively send probes to detect more link states according the passive monitoring result.

The following discussion of the paper is organized as follows. In Sect. 2, the general framework of our proposed adaptive fault monitoring solution will be introduced. In Sect. 3, the monitor placement scheme in our system, including the Integer Linear Programming (ILP) formulation, will be discussed. In Sect. 4, the numerical results of our framework will be presented. In Sect. 5, the flexibility of network design using our framework will be discussed. Section 6 summarizes the paper.

2 Adaptive Fault Monitoring

2.1 System Framework

In this section, we introduce the concept and architecture of our adaptive fault monitoring scheme for all-optical networks. Single link failure is assumed in this work, as it is the dominant scenario, and a link between two nodes represents a single fiber supporting bidirectional transmission.

In our system, a central control facility, that maintains the network resource information in a database, is assumed to be responsible for client request collection, routing path computation, and lightpath establishment by configuring the corresponding OXCs along the selected path, as well as failure detection and recovery. To achieve all these functions, a separated control plane is necessary. It can be implemented with either an out-of-fiber configuration or an in-fiber configuration, such as a dedicated wavelength channel. From the reliability point of view [7], it is more reasonable to adopt the out-of-fiber configuration, rather than the in-fiber configuration. For instance, the out-of-fiber configuration allows us to use different schemes to assure survivability, while the in-fiber configuration protection scheme should design mechanisms combining the data plane and the control plane. Besides, the out-of-fiber configuration supports the separation of the control plane and the data plane, which is crucial for maintaining network operation properly, in case of any failure in the data plane. In this work, we design a novel mechanism to perform fault localization in an all-optical network architecture, consisting of physically separated data and control planes, in which we try to minimize the capital costs to monitor the data plane network link state by fully utilizing the separated data plane and control plane architecture, combined with real-time data traffic.

Figure 1 shows the basic idea of our scheme, comparing both passive and proactive monitoring solutions. The overall process of our scheme comprises two phases. The first phase (Phase 1) is passive monitoring. One special feature of our passive monitoring system is to send passive monitoring probes according to the

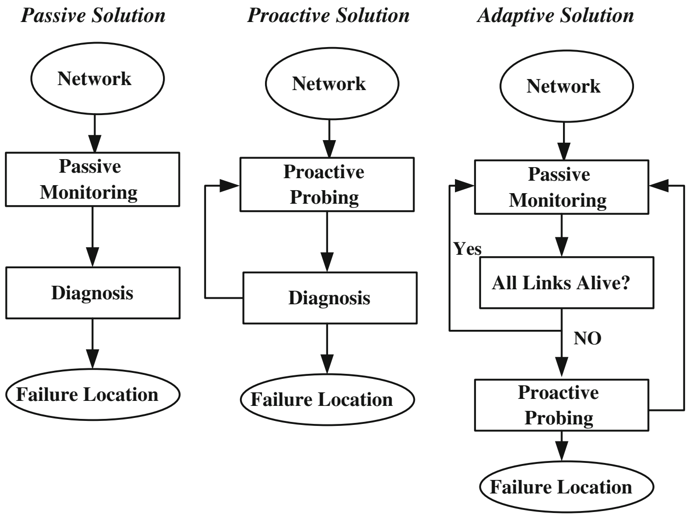


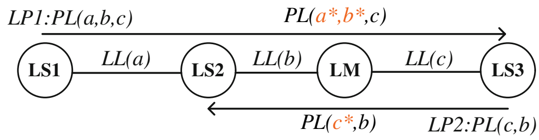
Fig. 1 Fault monitoring solutions: passive, proactive, and adaptive

data plane traffic condition and its routing method. Hence, the collected link state information (alive or failed) in the whole network merely depends on the previous traffic condition and its routing method, without any interruption to the data traffic. This feature will be further discussed in the later sections. After a short period of time, the network management plane checks all the received link state information and makes a decision whether it is necessary to execute Phase 2 or not. If Phase 2 is executed, the management plane triggers the designated source nodes to send probes to certain destination nodes to estimate the exact location of the failed link. In general, as one lightpath may be disrupted by a few possible fiber cuts in an all-optical network, Phase 1 aims to narrow down and sort out the possible failed links, while Phase 2 can further determine the actual location of the failure.

2.2 Phase 1: Passive Monitoring

The fault monitoring framework discussed in the previous section can be realized by a novel technique, namely label tracing monitoring (LTM). Ideally, there are three components in a LTM system: Label Source (LS), Link Label (LL) and Label Monitor (LM). LS injects link labels into the network, while different LLs are designed to denote different links in the network. LM is placed to gather labels in order to retrieve link state information. Figure 2 is a simple example to illustrate how these three components function. Each LS is capable of generating a path label (PL) that contains all link labels from source to destination in a lightpath. As shown in Fig. 2, a lightpath LP1 is to be established from LS1 to LS3. The central control facility receives such a request and configures the corresponding OXCs. It further asks LS1 to embed a path label (*a,b,c*) into the real time data traffic. This path label will be detected only at the LMs along the path, if any. In this example, the path label (*a,b,c*) is received at the LM. As this label is detected at the end of link *b*,

Fig. 2 An example to illustrate the label tracing monitoring (LTM) system



the system can only retrieve information that links a and b are alive (denoted as a^* and b^*), since the labels in a path label are ordered. Similarly, if another lightpath LP2 is set up from LS3 to LS2, only label c can be retrieved at the LM (denoted as c^*). If these two lightpaths are set up simultaneously, labels a^*, b^* and c^* will all be received at the LM, which indicates that no link failure happens. This procedure is regarded as Phase 1. It is passive, as no pre-designed probing scheme is involved. The probes are sent according to the traffic pattern that makes control costs low. The goal for Phase 1 is to fully utilize the randomness of the real traffic and thus reduce the management costs.

2.3 Phase 2: Proactive Probing

At a certain time checkpoint, the system will check the label information received at the LMs. If all link labels in the whole network are collected, it means all links are alive and thus Phase 2 will not be executed. Phase 2 is executed if there are one or more link labels missing.

As stated before, Phase 2 proactive probing is based on the result of the Phase 1. To avoid any disturbance to the data plane traffic, Phase 2 proactive probing tries to utilize the free wavelength resource in the network to do the link fault diagnosis. In other words, with the available wavelength resource, a set of LM locations and the missing link labels at the LM after Phase 1, a feasible probing algorithm is executed to detect the states of those links with missing link labels.

First of all, we introduce a graph model commonly used in a traffic grooming problem to represent the available wavelength resource in the whole network. The graph model has W planes, where W is the number of wavelengths supported by the network. Each plane, denoted as $\boldsymbol{G}_\lambda(\boldsymbol{V}_\lambda, \boldsymbol{E}_\lambda)$, corresponds to a particular wavelength λ and the nodes, $\boldsymbol{V}_\lambda$, in each plane correspond to the nodes in the physical topology. In each plane, there is an edge, $\boldsymbol{E}_\lambda$, between two nodes if a fiber link exists between the two nodes, and the relative wavelength is free in that fiber link. With this graph model information, we design a heuristic for network diagnosis. The basic idea is trying to find a set of proactive trails that could uniquely identify the fault that may happen to those links with missing link labels after Phase 1, by observing the instant link labels at the LMs. The proactive trail formation rule is similar to that in [8]. In our algorithm, a *trail*, which denotes a lightpath, ends at a certain LM on a particular wavelength plane. Nevertheless, due to the limited wavelength resources, there may not be enough trails that can localize any single link failure. To overcome this problem, we adopt a well studied concept, *shared risk link group* (*SRLG*) [9], to represent a group of links that cannot be distinguished by the current proactive trail formation. Ideally, there is only one member in each shared risk link group. The process of our algorithm is simple. After the greedy trail formation, we proactively send labels along these chosen paths. If no link failure is detected, we move to the next round of proactive trail formation, to cover those links with missing link labels. On the contrary, if a link failure is detected, a candidate failed link group would be uniquely identified. With the failed link group, the mesh topology fault localization problem has been transformed to a linear topology that is much easier to solve.

Before illustration of the proactive failure detection algorithm, we first discuss the principle of the formation of *SRLG*. In each wavelength plane λ, that is $\boldsymbol{G_\lambda}(\boldsymbol{V_\lambda}, \boldsymbol{E_\lambda})$, $\boldsymbol{LM_\lambda}$ is the set of LMs using wavelength λ. Those links whose link labels are not received in Phase 1, and are present distinctly in their own wavelength plane, are assigned to $\boldsymbol{h_\lambda}$, where $\boldsymbol{h_\lambda} \in \boldsymbol{H}$ for all λs. The algorithms, as illustrated in Fig. 3, are adopted to form a *monitoring trail* set, denoted as $\boldsymbol{I_\lambda}$, that covers the links in $\boldsymbol{h_\lambda}$. The breadth-first search algorithm is adopted, starting from each LM placed in $\boldsymbol{G_\lambda}(\boldsymbol{V_\lambda}, \boldsymbol{E_\lambda})$ until all the nodes connected to every LM are included in the same connected component. In graph theory, a connected component [10] of an undirected graph is a sub-graph in which any two vertices are connected to each other by paths, and which is connected to no additional vertices. Clearly, there are two possible situations at this stage. One includes only a single LM in the connected component, while the other contains multiple LMs. Thus, we have developed two heuristic forming policies, namely, *a trail formation policy for connected graph with single LM* and *a trail formation policy for connected graph with multiple LMs*, for trail formation under these two cases, respectively. Consequently, after all wavelength planes have been considered, an initial set of *SRLG,* denoted as $\boldsymbol{I}$, where $\boldsymbol{I_\lambda} \in \boldsymbol{I}$ for all λs, is formed as *proactive trails*.

Trail formation policy for connected graph with single label monitor (LM)

Step1: Reduce the original $G_\lambda(V_\lambda,E_\lambda)$ by eliminating those sub-graph component with no link$\in \boldsymbol{h}_\lambda$.
Step2: Using the obtained tables from the breadth-first searching algorithm in the previous step, starting from the only LM as the source,
if (the first link $\boldsymbol{l} \in \boldsymbol{h}_\lambda$ is found)
{ form the trail $\boldsymbol{t}$ connecting $\boldsymbol{l}$ to the only label monitor with the shortest path.
$\boldsymbol{h}_\lambda = \boldsymbol{h}_\lambda - \boldsymbol{l}$;
$\boldsymbol{I}_\lambda = \boldsymbol{I}_\lambda \cup \boldsymbol{t}$;
Go to **Step3**;
}
else go to **Step4;**
Step3: Delete the trail $\boldsymbol{t}$ from $G_\lambda(V_\lambda,E_\lambda)$, update the tables from breadth-first searching algorithm, and go back to **Step2**.
Step4: if ($\boldsymbol{h}_\lambda = \varnothing$) go to **Step5**
else { check whether if any links in $\boldsymbol{h}_\lambda$ can be connected to trails in $\boldsymbol{I}_\lambda$.
If yes, update $\boldsymbol{I}_\lambda$.
}
Step5: Return $\boldsymbol{I}_\lambda$
END

Trail Formation policy for connected graph with multiple label monitors (LMs)

Step1: Reduce the original $G_\lambda(V_\lambda,E_\lambda)$ by eliminating those sub-graph component with no link$\in h_\lambda$.

Step2: Let $LM'_\lambda = LM_\lambda$

Step3: Use the obtained tables from the breadth-first searching algorithm in the previous step. The search starts from every LM in LM'_λ over $G_\lambda(V_\lambda,E_\lambda)$. The nodes covered in every searching step have the same link distances to their respective source LMs.

if (the first link $l\in h_\lambda$ is found in the searching path of a label monitor $m\in LM'_\lambda$)

{ form the trail t connecting l to m with the shortest path;

If multiple links exist, break the tie by choosing the link that will block the minimal number of uncovered links in h_λ to be formed in potential trails.

$LM'_\lambda = LM'_\lambda - m$;

$h_\lambda = h_\lambda - l$;

$I_\lambda = I_\lambda \cup t$;

Delete the trail t from $G_\lambda(V_\lambda,E_\lambda)$, update the tables from breadth-first searching algorithm, and go back to **Step3**

}

else {

if ($h_\lambda = \varnothing$) go to **Step4**

else { $LM'_\lambda = LM_\lambda - LM'_\lambda$;

$LM_\lambda = LM'_\lambda$;

if ($LM'_\lambda \neq \varnothing$) go back to **Step3**;

else go to **Step4**

}

}

Step4: For those remaining links in h_λ, check whether if any one of them can be connected to trails in I_λ.

if (yes), update I_λ.

Step5: Return I_λ.

END

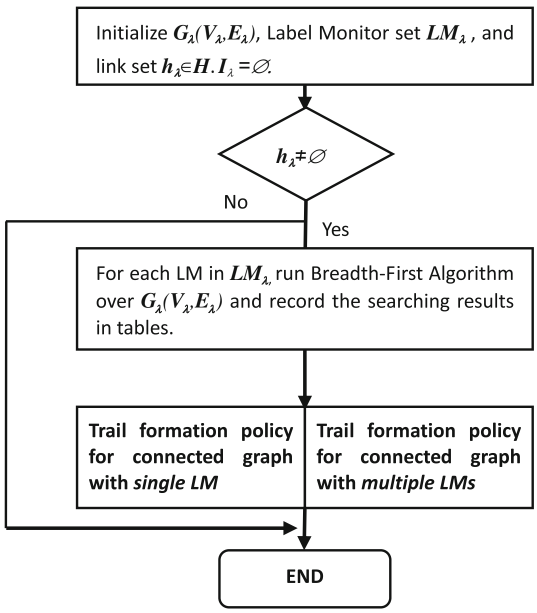


Fig. 3 Flowchart of the algorithm to form monitoring trail set in each wavelength plane

The initial set of *SRLG*, ***I***, is then used as the input for the *proactive failure detection algorithm*, as illustrated below.

Here are the notations in the *proactive failure detection algorithm*:

U All the links that are not detected alive in Phase 1
T The selected proactive trails
S Link set to be covered in ***T***
Wp The graph model showing free wavelength resource of the whole network
L As a subset of ***U***, denoting those links that are not covered by ***T*** while through which at least one lightpath can reach some LM in the current ***Wp***

Proactive Failure Detection Algorithm in Phase 2

Step 0: ***L=U*,** and ***S***=∅
  Let ***T=I***
  Update ***S***, ***L*** *and* ***Wp***

Step1: While ***L***≠∅
    Select a link ***l*** randomly from ***L***
    Compute a trail ***t*** passing through the nearest LM and ***l*** from ***Wp***
    T=T ∪ ***t***
    Update ***S***, ***L*** and ***Wp***
  End

Step2: if a set of *SLRG*, ***p*,** appear in the same set of trails
    Sort the link group in ***p*** into different groups if there is at least one different trail ***t*** between any two different groups in ***Wp***
    T=T ∪ ***t***
  End

Step3: Execute selected probing along the trails in ***T***;
  Return failure shared risk link group ***l****
  if ***l****=∅
    if (***U***≠∅)
    {
      Update ***U*** *and* ***I*** *according to* ***T***
      Return to **Step0**
    }
    else All links are alive
  else Failure_Location(*l)** //Failure_Location(.) is a function used to localize the exact location of fault within a shared risk link group//

To further illustrate this proactive failure detection algorithm, a simple example is shown in Fig. 4, where the six-node topology supports two wavelengths λ_1 and λ_2. In this example, two lightpaths are established in Phase 1, due to the real traffic requests. One is from LS5 to LM generating PL(*c*) on wavelength λ_1, while the other one is sourced at LS3 and sends PL(*e*, *d*) to LM on wavelength λ_2. As a result, the LM receives link label LL{*c*,*e*,*d*} during Phase 1 and regards links *c*, *e*, *d* as currently active. At a certain time, the control plane checks the link label information at the LM to find that link labels, namely, *a*, *b* and *f*, have not yet

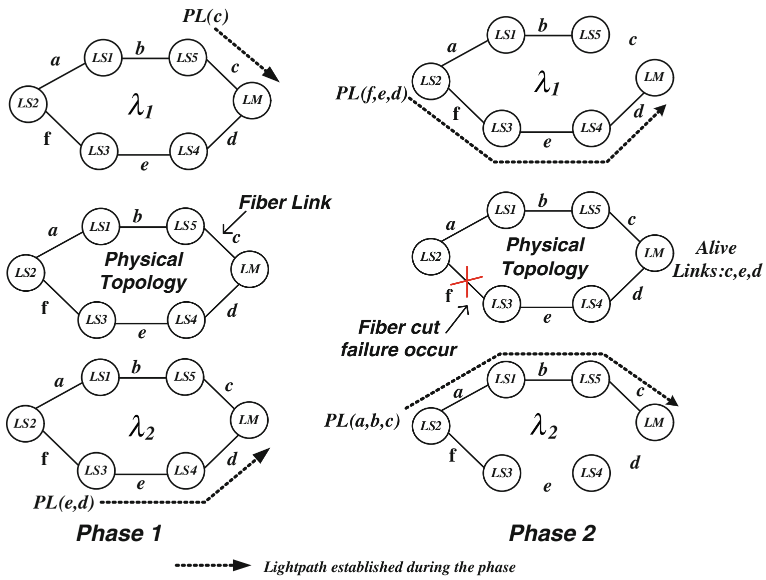


Fig. 4 An example to illustrate the procedures of Phase 2

received. Thus, we have $\boldsymbol{U} = \{a,b,f\}$ and $\boldsymbol{L} = \boldsymbol{U}$. To completely diagnose all network link states, Phase 2, the proactive failure detection algorithm, is executed according to the results of Phase 1 and the current network wavelength resource. To accomplish that, we randomly pick a link from those links with missing link labels, including a, b and f in $\boldsymbol{L}$. We first assume that f is selected, without loss of generality. Please note that the order in which the links are selected does not affect our final failure localization results. Then a trail passing through the nearest LM and f is computed in the graph model. The trail is added to set $\boldsymbol{T}$ and those links with missing link labels along the trail are added to $\boldsymbol{S}$. In the above case, one and the only one of the possible trail (f,e,d) on wavelength λ_1 is put into $\boldsymbol{T}$, while link f is added to $\boldsymbol{S}$. Now, updating the graph model means that the lightpath from LS2 to LM on λ_1 plane is removed. Here, since link a and link b are still not covered and there are only one trail, passing through both of them, in the graph model, we will group the two links as a unique link group that can be treated as a virtual link, corresponding the updating set $\boldsymbol{I}$ in Step 3 of the algorithm. With this transformation, there is only one *SRLG* including physical links a and b in $\boldsymbol{L}$, and therefore, the same procedure is employed to this link class, as happened to link f. Thus, another trail (a,b,c) on wavelength λ_2 is added to $\boldsymbol{T}$. This greedy proactive trail formation policy is performed until $\boldsymbol{L}$ is empty. In the above simple example, the two trails, (f,e,d) and (a,b,c), can already locate the fiber failure at link f.

Obviously, for more complicated cases, those covered *SRLGs* appeared in the same set of trails may arise after the initial trail formation procedure. To address this problem, if the unused wavelength resource is available, the algorithm will add new trails to distinguish these *SRLGs*. Otherwise, these *SRLGs* will be grouped as a larger SRLG. With this greedy proactive probing along the selected trails, we are able to uniquely identify a failed link group if a failure happens to those links covered in proactive trail set ***T***. If no failure happens, the algorithm will recursively update and probe the remaining un-covered links until a failure is identified or all the link labels are received.

2.4 Control Plane Design and Timeline Analysis

In this section, the design of the control plane will be discussed. Two important concepts, namely, Reliable Label (RL) and Reliable Time (RT), are defined to facilitate our discussions.

Reliable Label (***RL***) *The labels stored at LMs which indicate the respective links are alive with probability p*

Reliable Time (***RT***) *For each collected label at LMs, the time interval within which the label is treated as an RL from the instant the label is received*

Due to the intuitive fact that the occurrence probability of a link failure is getting larger when the respective link label is missing for a longer period of time, a threshold value for RT is selected to put a probing effort on the links with a larger failure probability. The probability is determined by real network conditions and is monotonically decreasing with RT. Basically, RT is equivalent to the assumption used in [5] that the link state will not change during the proactive probing process, that is, within a certain time interval after the moment the label is detected. Thus, the previously detected link information will be reliable in the following proactive time. In the design of the control plane, network monitoring is performed in time intervals of ΔT, as illustrated in Fig. 5. Within each time interval, a time instant is set as Check Point (CP), at which the system will check the RL list at LMs and make decisions for the execution of Phase 2. Phase 2 will be performed in the following time period between CP and the end of the current time interval ΔT, so as to limit the fault recovery time. At the end of Phase 2, fault notification will be issued if there are still any links with missing link labels. The time interval between two consecutive CPs is also equal to ΔT. In case that Phase 1 and Phase 2 are performed in parallel, Phase 1 is collecting labels all the time, while Phase 2 periodically sends probes according to the RL list at each CP. Moreover, the longest time interval between failure occurrence and failure detection is bounded in our scheme, as explained in lemma 1.

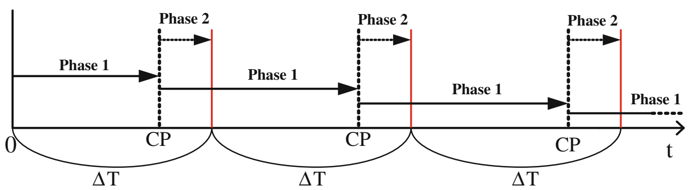


Fig. 5 Timeline of the proposed adaptive monitoring scheme

Lemma 1 *For any network topology, if all the nodes are LS, with at least one LM, the single link failure localization time is bounded by $RT + \Delta T$*

Proof In our fault localization scheme, the missing link label after Phase 2 corresponds to the failed link. From the start of each ΔT period, two possible situations for each link may occur. First, if there is at least one passive probe (lightpath established in Phase 1) passing through it before being received at some LMs, it will be detected at the first CP after its latest RT ends. Thus, if a link fails after its label has been collected at some LMs, the worst case would be that its RT ends just after a CP and this link is not included in any *shared risk link groups* in the following Phase 2 detection. In that case, an additional ΔT would be needed. Therefore, the worst case fault localization time would be RT + ΔT. On the other hand, if there is no passive probe passing through certain links or a failure occurred before the passive probe is sent, the worst case fault localization time would be ΔT, since Phase 2 will be executed to detect the respective label for the failed link.

Lemma 1 proves that our fault detection and localization scheme is reliable and complete. By choosing RT and ΔT flexibly, our scheme can be adjusted adaptively according to different network link failure models and QoS specifications, in terms of reliability. For instance, the checktime interval ΔT can be reduced in a network that requires a short recovery time. Hence, the system can locate the possible link failure within a shorter period. On the other hand, if the network operator is more concerned about the fault management cost than the recovery time, increasing ΔT can reduce the average amount of control messages sent in a certain period. Moreover, if fiber links in a particular network are exposed in a hostile or fast changing environment, the RT can be set to a smaller value. Besides, our scheme performs better when the real traffic load is getting higher, as more link labels will be received at the LMs in Phase 1. Hence, the network can improve the QoS, in terms of reliability, at about the same management cost.

2.5 Physical Label and Monitor Implementation Suggestions

A path label will be sent out when a lightpath has been established and the label will travel along the lightpath along with the data. As the label is used to denote the path information of the lightpath on a particular wavelength, there will be several lightpaths going through the input port of the label monitor on different wavelengths. In order to gather the link label information from different lightpaths, as much as possible, it will be desirable to have a low-cost label monitor that can monitor the link labels on several different wavelengths, simultaneously. For example, subcarrier multiplexing may be employed to support multiple different labels, each of which are carried on a distinct subcarrier frequency. On the other hand, code-division multiple access (CDMA) is also a feasible alternative, as proposed in [11]. It integrates a direct-sequence CDMA (DS-CDMA) technique with a complementary constant weigh code (CCWC). DS-CDMA is used to multiplex different link labels at baseband frequencies and CCWC is used to overlay the low-speed label into the high-speed payload. Instead of CCWC, Manchester coding or coded marked inversion (CMI) coding may also be used to encode the payload, such that there is a spectral null at DC, and thus enabling the insertion of the low-speed label.

3 Placement of Label Monitor (LM)

In the previous sections, we have illustrated the architecture and the principles of our adaptive fault monitoring system in an all-optical network. In this section, the LM placement problem is discussed. It is shown that our scheme can be embedded into optical networks without any disturbance incurred to the data plane control. Moreover, among the three components (LS, LM and LL), the deployment cost of LM dominates the monitoring system cost. Therefore, it is highly desirable to minimize the number of the required label monitors and monitoring ports.

3.1 Problem Formation

Given a network topology $\boldsymbol{G}(\boldsymbol{V},\boldsymbol{E})$, $\boldsymbol{V}$ represents the set of all nodes and $\boldsymbol{E}$ denotes the set of all links. Also, a bidirectional fiber link is assumed. In our adaptive scheme, each link of the network uses a unique value as its link label (LL). As discussed in Sect. 2, we simply assume that every node of the network is a label source (LS), since LS actually introduces negligible costs compared with the cost of LM. Our objective is to place the minimum number of monitors in the network, subject to the constraint that all link labels can be collected at LMs in Phase 1, if ΔT is long enough. To realize this objective, we present a simple placement solution formulated as an integer linear programming (ILP).

3.2 Minimum LM Placement

As discussed in Sects. 1 and 2, where and how the probes are sent in Phase 1 are determined by the data plane traffic and its routing method. Without loss of generality, the most common fixed-alternative k-shortest paths routing for the data plane is chosen, where k is set to be 2. Note that, in this ILP formation, no actual traffic model parameters, such as the time interval distribution between two consecutive requests and holding time of requests are involved. We simply assume the traffic demands are uniformly distributed to all node pairs, indicating that all the k pre-computed paths will be used. In terms of future backbone all-optical networks, we believe the traffic nature would be increasingly dynamic. Yet, all other wavelength routing algorithms could be used and all possible traffic models could be adopted. The following ILP formulation is an example, for illustration.

Minimize: $\sum_{i \in V} NiDi$

Subject to: $\sum_{i \in V} CijDi \geq 1 \quad j \in E$

Notations:

V The node set of the topology

E The link set of the topology

N_i Degree of node i

D_i Placement binary variable. $D_i = 1$ means that node i is selected to place a LM; otherwise, $D_i = 0$

C_{ij} Binary input of the network. $C_{ij} = 1$ indicates that link j is included at least once by those two shortest paths between node i and all the other nodes

In general, our objective is to minimize the overall number of LM ports subjected to the constraint that all the link labels could be possibly received at any LM in Phase 1. Variable D_i represents the decision whether node i is selected to be an LM. Since each LM should receive and monitor all the links attached to it, N_i, the degree of node i, is added as weight to include this property in the monitoring cost objective function, which is to minimize. C_{ij} is a binary input value, indicating that if link j is included at least once, by those two shortest paths, between node i and all the other nodes. It can be achieved by offline computing the hop distance based 2 shortest paths between every node pair, using the Dijkstra algorithm. We would like to point out that the traffic from any of the other nodes to this LM is a broader concept because LM can collect labels that pass through it, in addition to those destined at this LM.

In the above ILP formation input, k could be chosen to be other values. Larger k usually means a smaller number of LMs required. Thus, the link status information collected in Phase 1 is reduced and this may increase the burden of Phase 2. Besides, the LM placement method could consider other alternative routing algorithms, such as adaptive dynamic routing, in which a routing decision is made on the fly depending on the network wavelength resource. However, as there is no fixed route between the source and the destination in adaptive dynamic routing, a heuristic would be required to choose those hub nodes, for instance, via ranking the nodes with their node degrees.

4 Performance Study and Analysis

In this section, we present the numerical results of our proposed framework and the comparison with M-Trail, which is a popular previously proposed solution for single link failure localization in all-optical networks.

4.1 ILP Formulation Results over Random Topology

We have applied our ILP formulation to a number of randomly generated topologies to evaluate the performance of our LM placement scheme. The network model was implemented using C/C++ with a free library, called *ip_solve*, included in Microsoft Visio Studio 2005. In the above ILP formation input, the value of k was set to be 2. The traffic demands were uniformly distributed to all node pairs. For a different number of nodes, we start our placement from a ring topology (average node degree is 2) until a fully mesh network [4] is formed. For each average node degree value, denoted as *AND*, we randomly generated 100 topologies to get the average number of the monitor ports, which is set to be the cost value of our results,

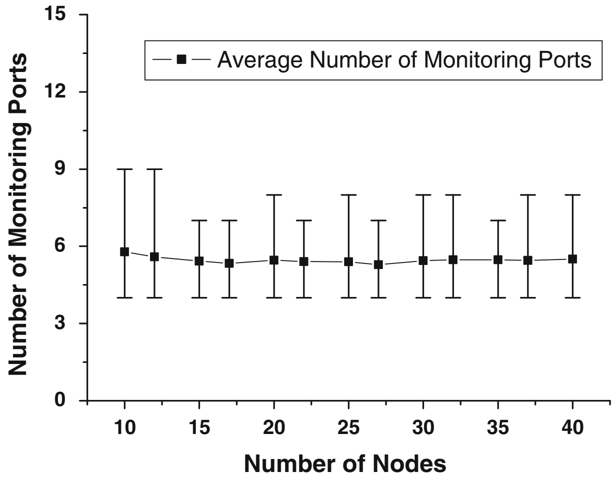


Fig. 6 Monitor cost (number of monitoring ports) versus node number with average node degree 4

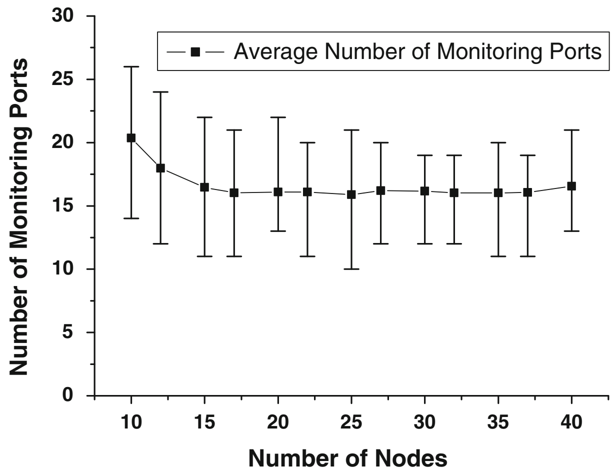


Fig. 7 Monitor cost (number of monitoring ports) versus node number with average node degree 6

while the maximum monitor ports and the minimum monitor ports in those 100 topologies, are also presented. Figs. 6 and 7 show the monitor costs of different network sizes to realize the Phase 1 function under *AND* values of 4 and 6. From Figs. 6 and 7, we have found that the monitor cost does not increase if the number of nodes and links maintaining the same average node degree. It is a very useful property, indicating that our adaptive scheme has great scalability, in terms of monitoring costs. The reason is also intuitive. As the number of links to be monitored increases, the number of LSs also increases by approximately the same ratio (*AND*). As the number of network nodes increases, the overall possible lightpath in the network will also increase. Thus, those new links can be covered easily by the new possible lightpath without extra monitoring costs. Figure 8 puts the test results on networks with average node degree of 2, 4 and 6 in the same figure. Figures 9, 10, 11, 12 and 13 show the monitor cost versus *AND*, for a different number of nodes, together with the maximum and the minimum test results, in form of error bars, from the 100 topologies used. Figure 14 puts the test results on networks with node numbers of 6, 10, 14, 20 and 24, in the same figure. It is clear that our adaptive system cost increases with network *AND*, but not with the

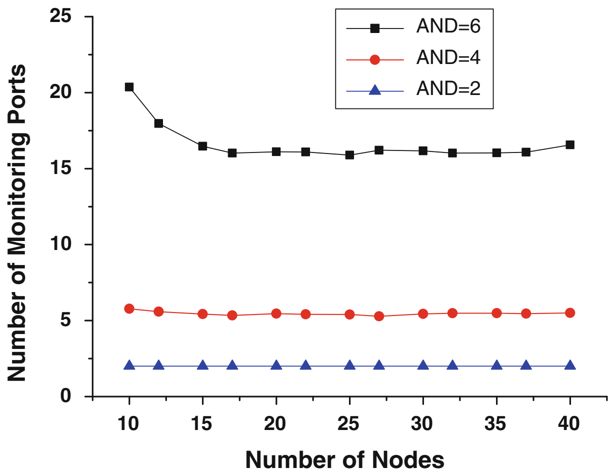


Fig. 8 Monitor cost (number of monitoring ports) versus node number with same average node degree (*AND*)

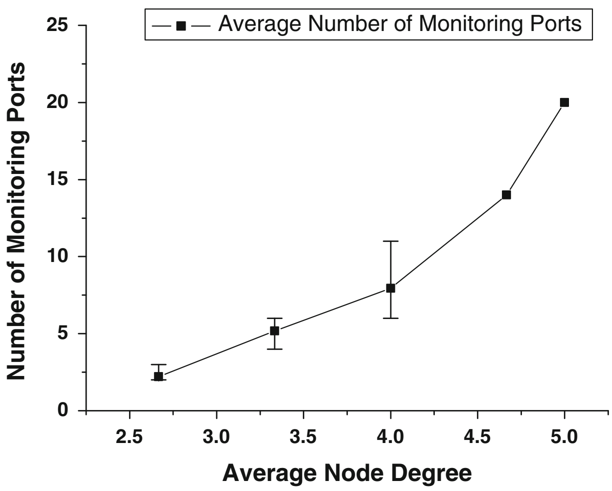


Fig. 9 Monitor cost (number of monitoring ports) versus *AND* with node number 6

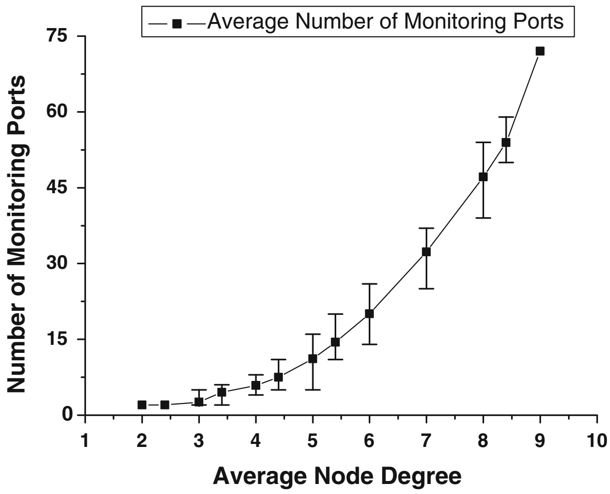


Fig. 10 Monitor cost (number of monitoring ports) versus *AND* with node number 10

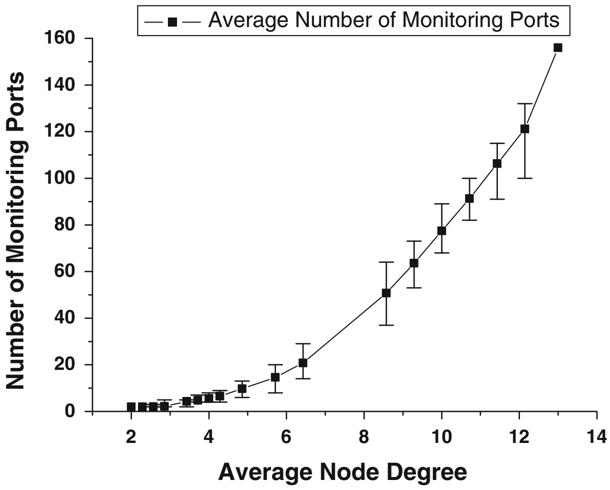


Fig. 11 Monitor cost (number of monitoring ports) versus *AND* with node number 14

number of nodes. Moreover, the monitor cost is shown to be relatively low in the *AND* interval between 2 and 5, which is within the *AND* interval in real networks.

4.2 Monitoring Cost Comparison with M-Trail and M-Cycle Design

One major advantage of our adaptive monitoring framework, as compared with M-Trail and M-Cycle, is that almost no dedicated bandwidth is consumed for

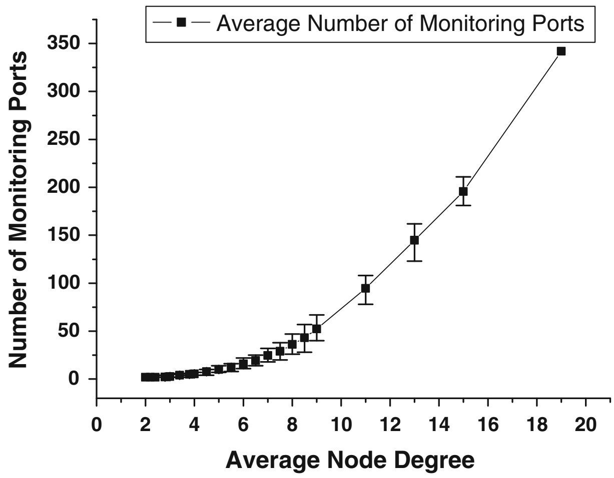


Fig. 12 Monitor cost (number of monitoring ports) versus *AND* with node number 20

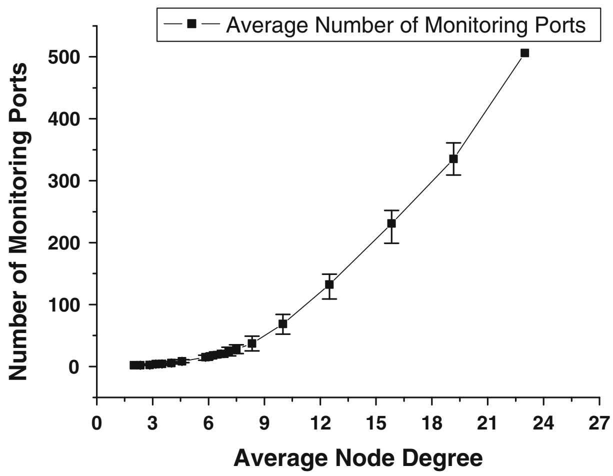


Fig. 13 Monitor cost (number of monitoring ports) versus *AND* with node number 24

monitoring, since our adaptive scheme uses the free wavelengths of the data plane, while M-Trail and M-Cycle may require a significant number of dedicated monitoring channels, as the network dimension increases. Furthermore, the number of monitors required in our framework, M-Trail and M-Cycle are compared. Table 1 shows the results of our monitor placement, together with the respective data extracted from the original M-Cycle design and M-Trail design [6, 12], for comparison. Several commonly used topologies, as shown in Fig. 15, are considered.

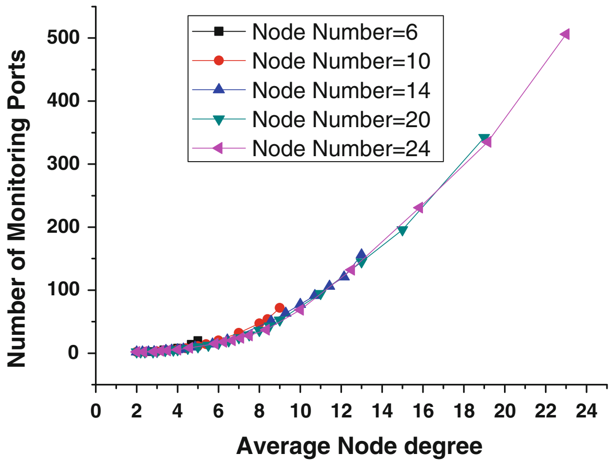


Fig. 14 Monitor cost (number of monitoring ports) versus *AND* with same node number

Table 1 Monitor Number Comparison with M-Cycle [6] and M-Trail [12]

Network topology	M-Cycle	M-Trail	Proposed Adaptive Scheme
ARPA2	20 cycles	11 trails	2 ports
SmallNet	13 cycles	6 trails	7 ports
NSFNET	10 cycles	Information not available*	4 ports
Bellcore	16 cycles	Information not available*	5 ports

* The authors are not able to find the references on the number of trails in those network topologies from available resources

5 Discussion

In the previous sections, we present a hybrid and adaptive fault localization framework that works by combining passive monitoring on real-time data traffic and proactive monitoring solutions together. By monitoring real-time data traffic in Phase 1, we can reduce the number of proactive probes in Phase 2. Moreover, the fast changing real-time data traffic in optical networks will reduce the required number of fault monitors to be placed in the network, compared with previous passive monitoring schemes, which utilize static monitoring traces or routes, since a large amount of link states are monitored by a single label monitor in our scheme. In addition, it is noticed that the number of monitors to be placed in the network is proportional to the average node degree of the optical network. Hence, any insertion of network nodes will not increase the monitoring costs much, as long as the average node degree of the optical network is maintained. Figure 16 shows an example of how the monitoring solution changes when the network topology changes, i.e., addition or deletion of network nodes. In Fig. 16(a), M-trail scheme

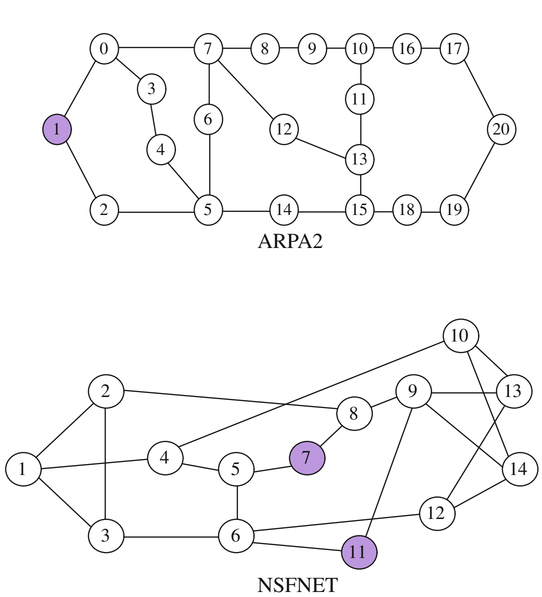


Fig. 15 Several commonly used topologies for monitor number comparison (*shaded nodes* denote monitoring locations using proposed scheme)

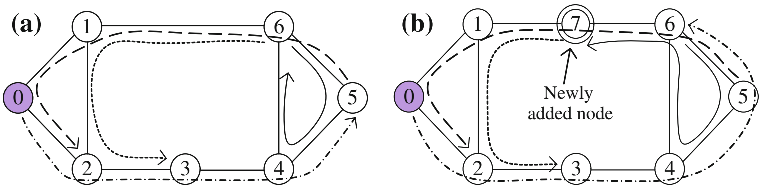


Fig. 16 Monitoring solutions updated using different monitoring schemes(proposed scheme and M-Trail) when network topology changes (**a**) the original network (**b**) the new network (*shaded nodes* denote monitoring locations using proposed scheme and *arrows* denote monitoring trails using M-Trail scheme)

requires 4 trails marked by the different styles arrows, while our scheme requires the placement of the monitor at network node 0. When network node 7 is inserted to the network, as shown in Fig. 16(b), although the total number of trails can be kept constant, the trail configuration has to be changed. This leads to the change of the placement of monitors, lasers, and even the connection pattern of the supervisory channel at the network nodes. Nevertheless, our scheme requires no additional modification, in this example. Further analysis will be conducted to examine the robustness of our proposed scheme under network topology changes. Besides, M-trail's ILP calculation [12] requires a large number of constraints that increases the complexity of computation. The situation would get even worse for a network with larger dimensions. Due to the ease of the computation of the monitor location of our scheme, much less computation time is required whenever there are changes in the network topology.

6 Summary

We propose a novel and practical adaptive fault monitoring scheme in all-optical networks based on the label tracing monitoring (LTM) method. A simple, yet effective monitor placement method using ILP is presented. Our results show that our adaptive fault localization scheme has great scalability in terms of the lowest number of fault monitors required. Besides, our scheme performs better, in terms of design flexibility and minimal additional dedicated monitoring bandwidth, than the common passive monitoring solutions.

References

1. Alexander, S.B., Bondurant, R.S., Byrne, D., Chan, V.W.S., Finn, S.G., Gallager, R., Glance, B.S., Haus, H.A., Hu, P., Jain, R., Kaminow, I.P., Karol, M., Kennedy, R.S., Kirby, A., Le, H.Q., Saleh, A.A.M., Schofield, B.A., Shapiro, J.H., Shankaranarayanan, N.K., Thomas, R.E., Williamson, R.C., Wilson, R.W.: A precompetitive consortium on wide-band all-optical networks. IEEE/OSA. J. Lightwave Technol. **11**(5/6), 714–735 (1993)
2. Ramaswami, R., Sivarajan, K.N.: Optical networks: A practical perspective, 2nd edition. Morgan Kaufmann Publishers, Los Altos, CA (2002)
3. Maeda, M.W.: Management and control of transparent optical networks. IEEE. J. Sel. Areas Commun. **16**(7), 1008–1023 (1998)
4. Tapolcai, J., Wu, B., Ho, P.-H: On monitoring and failure localization in mesh all-optical networks. In Proceedings of IEEE INFOCOM 2009, 1008–1016 (2009)
5. Wen, Y.G., Chan, V.W.S., Zheng, L.Z.: Efficient fault diagnosis algorithms for all-optical WDM networks with probabilistic link failures, IEEE/OSA J Lightwave Technol. **23**(10), 3358–3371 (2005) (invited paper)
6. Zeng, H., Huang, C., Vukovic, A.: A novel fault detection and localization scheme for mesh all-optical networks based on monitoring-cycles. Photon Netw. Commun. **11**(3), 277–286 (2006)
7. Jajszczyk, A., Rozycki, P: Recovery of the control plane after failures in ASON/GMPLS Networks, IEEE Network. **20**(1) (2006)
8. Ahuja, S., Ramasubramanian, S., Krunz, M.: Single link failure detection in all-optical networks using monitoring cycles and paths. IEEE/ACM. Trans. Network. **17**(4), 1080–1093 (2009)
9. Strand, J., Chiu, A., Tkach, R.: Issues for routing in the optical layer. IEEE. Commun. Mag. **39**, 81–87 (2001)
10. http://en.wikipedia.org/wiki/Connected_component_(graph_theory)
11. Vaishampayan, V.A., Feuer, M.D.: An overlay architecture for managing lightpaths in optically routed networks. IEEE. Trans. Commun. **53**(10), 1729–1737 (2005)
12. Wu, B., Ho, P.-H., Yeung, K.L.: Monitoring trail: on fast link failure localization in all-optical WDM mesh networks. IEEE/OSA J. Lightwave Technol. **27**(18), 4175–4185 (2009)

Author Biographies

Dong Shen received the B.S. and M.Phil. degrees, both in Information Engineering, from Zhejiang University, Hangzhou, China, in 2009, and The Chinese University of Hong Kong, Hong Kong, in 2011, respectively. His current research interests include fault management and resource optimization in optical networks.

Kam-Hon Tse received the B.Eng. degree in Information Engineering from The Chinese University of Hong Kong, Hong Kong, in 2008. He is currently working toward the Ph.D. degree in the same department. His current research interests include high-speed all-optical signal processing, and optical packet switching.

Chun-Kit Chan received the B.Eng., M.Phil., and Ph.D. degrees from The Chinese University of Hong Kong, Hong Kong. He was a Member of Technical Staff at Bell Laboratories, Lucent Technologies, Holmdel, NJ USA. Currently, he is an Associate Professor at the Department of Information Engineering, The Chinese University of Hong Kong. His main research interests include optical metro/access network design and high-speed optical signal processing.