Computer Communications 32 (2009) 777–781

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

A new algorithm based on auxiliary virtual topology for sub-path protection in WDM optical networks ☆

Xingwei Wang, Lei Guo *, Xuekui Wang, Yu Zhang, Xiaobing Zheng, Weigang Hou, Hongming Li, Hongpeng Wang

College of Information Science and Engineering, Northeastern University, P.O. Box 365, Shenyang 110004, China

ARTICLE INFO

Article history:
Received 19 July 2008
Received in revised form 30 November 2008
Accepted 3 December 2008
Available online 24 December 2008

Keywords:
Optical networks
Virtual topology
Sub-path protection
Protection-switching time
Time complexity

ABSTRACT

This paper proposes a new survivable algorithm named sub-path protection based on auxiliary virtual topology (SPAVT) to tolerate the single-link failure in WDM optical networks. First, according to the protection-switching time constraint, SPAVT searches multiple pairs of primary and backup paths for each node pair in the network by the off-line manner, and then map these paths to the virtual topology. When a connection request arrives, SPAVT only needs to run one time of the Dijkstra's algorithm to search a virtual route in virtual topology, where the route may consist of multiple pairs of sub-paths, to meet the protection-switching time constraint. Then, according to the shared resources policy, SPAVT chooses an optimal pair of sub-paths. Simulation results show that SPAVT has smaller blocking probability and lower time complexity than conventional algorithms.

1. Introduction

In wavelength-division-multiplexing (WDM) optical network, since each wavelength channel can provide very high transmission rate, the failure of fiber links may cause a lot of data loss. Therefore, survivability in WDM networks is more important [1]. In recent years, many papers have studied the survivable mechanism, in which the general idea is to take account the protection mechanism into the optical layer. According to the state of backup resources, the protection mechanism can be divided into dedicated protection, shared protection and mixed-shared protection [2–4]. According to the size of protection areas, the protection mechanism also can be divided into link protection, segment protection [5–8] and path protection [2–4].

In order to protect the single-link failure, the literature [2] proposed shared-path protection (SPP) algorithm, which searches one pair of a primary path and a link-disjoint backup path for each connection request. For any two connection requests, if their primary paths are link-disjoint, their corresponding backup paths can share the backup resources on the common fiber links. Although SPP can provide better resource utilization ratio and lower blocking probability than other shared protection algorithms, it does not consider guaranteeing the protection-switching time. The protection-switching time is an important factor in network performance and can not be ignored. The shorter the protection-switching time is, the sooner the restoration from the failures is.

In order to solve the problem of protection-switching time constraint, by integrating the merit of link protection and path protection, the literature [5–8] proposed the shared-sub-path protection (SSP) algorithm. According to the protection-switching time constraint, SSP first searches a primary path for each connection request, and then it separates the primary path into several equal length sub-primary paths and follows to compute a link-disjoint sub-backup path for each sub-path. Since the primary path will be divided into several sub-primary paths, SSP may run multiple times of the routing algorithm (e.g., Dijkstra's algorithm) to compute several sub-backup paths, and then the route computation time will increase and the algorithm time complexity will be high.

In order to improve the problem in previous studies, in this paper, we propose a new survivable algorithm named sub-path protection based on auxiliary virtual topology (SPAVT) to tolerate the single-link failure in WDM optical networks. According to the protection-switching time constraint, SPAVT first searches multiple pairs of working and backup paths for each node pair in the network by the off-line manner, and then map these paths to the virtual topology. When a connection request arrives, SPAVT only needs to

☆ The preliminary work of this paper was presented at the International Conference on Convergence and hybrid Information Technology (ICCIT) 2008. This work was supported in part by the National Natural Science Foundation of China (Nos. 60802023, 60673159, 70671020), the National High-Tech Research and Development Plan of China (No. 2007AA041201), the Key Project of Chinese Ministry of Education (No. 108040), the Specialized Research Fund for the Doctoral Program of Higher Education (Nos. 20070145096, 20070145017, 20060145012), and the Program for New Century Excellent Talents in University.

* Corresponding author. Tel./fax: +86 24 83687758.
E-mail address: haveball@yahoo.com (L. Guo).

0140-3664/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.comcom.2008.12.007

run one time of the Dijkstra's algorithm to search a virtual route in the virtual topology, where the route may consist of multiple pairs of sub-paths, to meet the protection-switching time constraint. Then, according to the shared resources policy, SPAVT chooses an optimal pair of sub-paths. Therefore, compared with previous algorithms, with the protection-switching time consideration SPAVT has smaller blocking probability than SPP; with the routing computation time consideration, SPAVT has lower time complexity than SSP.

The rest of this paper is organized as follows: Section 2 presents the network model, the protection-switching time, and the auxiliary virtual topology; Section 3 proposes the heuristic steps of SPAVT; Section 4 presents the simulation results and analysis; Section 5 concludes this paper.

2. Problem statement

2.1. Network model

The given network is denotes as $G = (N,L,W)$, where N is the set of nodes, $L$is the set of links and W is the set of available wavelengths on each fiber. The routing algorithm is Dijkstra's algorithm. Some important notations are defined as follows:

$l(\in L)$: bidirectional fiber link in the network
C_l: basic cost of link l, and it is determined by many factors, such as the physical length of link, the constructing cost of link, and so on
$|\Omega|$: number of the elements in set Ω
$K_p^{x,y}$: set of primary paths between node pair (x,y)
$K_p^{x,y}[i]$: The ith primary path in $K_p^{x,y}$, $1 \leqslant i \leqslant |K_p^{x,y}|$
$K_{b,i}^{x,y}$: set of backup paths which are link-disjoint with $K_p^{x,y}[i]$ between node pair (x,y)
$K_{b,i}^{x,y}[j]$: the jth backup path in $K_{b,i}^{x,y}[j]$, $1 \leqslant j \leqslant |K_{b,i}^{x,y}|$
G': virtual topology corresponding to G
$R(a,b)$: request with source node a and destination node b
(a',b'): virtual node pair in virtual topology, corresponding to node pair (a,b).

2.2. Protection-switching time

The time taken from the instant a link fails to the instant the backup path of a connection traversing the failed link is enabled is defined to be the protection-switching time. According to [1], we assume that the message-processing time D at a node is 10 µs, the propagation delay P on each link is 400 µs, the time X to configure, test and set up an OXC is 10 µs, the time$F$ to detect a link failure is 10 µs, n is the number of hops from the link l source to the source node of the connection (assume link l is the failed link on primary path or sub-path), and h is the number of hops of the backup or sub-backup path. Therefore, the protection-switching time t_k for sub-path k is written as:

$$t_k = F + n \times P + (n+1) \times D + 2 \times h \times P + 2 \times (h+1) \times D + (h+1) \times X \quad (1)$$

It is clearly shown that if the length of sub-path is long, the values of n and h will be big. Therefore, the protection-switching time will be long.

2.3. Protection on auxiliary virtual topology

The proposed SPAVT algorithm in this paper is a special case of sub-path protection under the protection-switching time constraint. First, SPAVT computes k primary paths by the k-shortest path algorithm for each node pair (m,n) according to the protection-switching time constraint, and then form the set $K_p^{m,n}$. Second, SPAVT computes k link-disjoint backup paths by the k-shortest path algorithm for each primary path $K_p^{m,m}[i](1 \leqslant i \leqslant |K_p^{m,n}|)$ meanwhile satisfying the protection-switching time constraint, and then form the set $K_{b,i}^{m,n}$. If $|K_p^{m,n}| \geqslant 1$ and $|K_{b,i}^{m,n}| \geqslant 1 (1 \leqslant i \leqslant |K_p^{m,n}|)$, it means that there exist the available pairs of primary and backup paths between node pair (m,n). Therefore, we can add a link between (m',n') in the virtual topology G'.

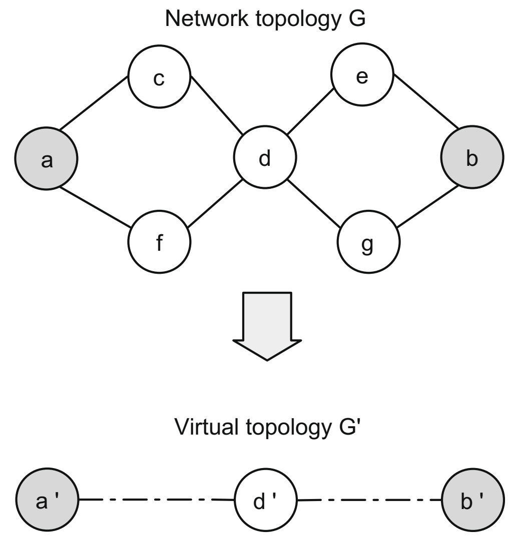

Fig. 1. Illustration of network topology G and virtual topology G'.

As shown in Fig. 1, we assume that there exist a primary path a–c–d and a backup path a–f–d which satisfy the protection-switching time constraint between the node pair (a,d), and there exist a primary path d–e–b and a backup path d–g–b which satisfy the protection-switching time constraint between the node pair (d,b), so that we can, respectively, add a link between node pair (a',d') and node pair (d',b') in the virtual topology. When a connection request $R(a,b)$ arrives, we only need to search a virtual route $a' - d' - b'$, on which the virtual links $a' - d'$ and $d' - b'$ both can meet the protection-switching time constraint.

Compared with conventional algorithms, SPAVT algorithm has the following advantages:

(1) Through searching routes in advance according to the protection-switching time constraint, SPAVT can find multiple pairs of feasible primary and backup paths for each node pair, and can solve the problem of blocking connection requests due to un-satisfying the protection-switching time constraint in conventional SPP algorithm. In Fig. 1, for $R(a,b)$, although SPP may find a primary path a–c–d–e–b and a backup path a–f–d–g–b, $R(a,b)$ will be blocked because the lengths of its primary and backup paths are too long to meet the protection-switching time constraint. However, SPAVT can find a virtual route $a' - d' - b'$ in virtual topology G', and the virtual links $a' - d'$ and $d' - b'$ both can meet the protection-switching time constraint. Therefore, under the protection-switching time constraint, SPAVT can perform better than conventional SPP in blocking probability.
(2) In addition, for a given physical network topology, searching routes in advance can be done by one time of the offline manner. Therefore, in virtual topology, SPAVT only need to run one time of the Dijkstra's algorithm to search the pairs of primary and backup paths for each connection request to meet the protection-switching time constraint,

and then it reduces the running times of the Dijkstra's algorithm comparing with conventional SSP. Thus, it is obvious that SPAVT can reduce the consumed time for routing computation. In Fig. 1, for $R(a,b)$ SSP will first run one time of the Dijkstra's algorithm to search a primary path a–c–d–e–b, and then it will run one time of the Dijkstra's algorithm to search a backup path a–f–d–g–b. Since the lengths of the primary and backup paths are too long to meet the protection-switching time constraint, SSP will first separate the primary path into two sub-paths a–c–d and d–e–b at node d, and then it will run one time of the Dijkstra's algorithm to find the sub-backup paths a–f–d and d–g–b for the two sub-paths, respectively. Therefore, SSP will run four times of Dijkstra's algorithm to search multiple pairs of primary and backup paths to meet the protection-switching time constraint. However, since the pairs of primary and backup paths have been found in advance by the off-line manner and they also have been mapped to the virtual topology, SPAVT only needs to run one time of the Dijkstra's algorithm to find a virtual route a'–d'–b'. It is clearly shown that the time complexity of SPAVT is lower than that of SSP.

3. Algorithm description

The heuristic steps of the proposed SPAVT are presented as follows:

Step 1: Initialize resources and costs in the network. Under the protection-switching time constraint, compute k primary paths by the k-shortest path algorithm for each node pairs (x, y) to form set $K_p^{x,y}$, and then compute k backup paths by the k-shortest path algorithm for each primary path $K_p^{x,y}[i](1 \leqslant i \leqslant |K_p^{x,y}[i]|)$to form the $K_{b,i}^{x,y}$, in which the backup paths are link-disjoint with the corresponding primary path by removing these links traversed by primary path before computing backup paths. Then, map network topology G to virtual topology G'.

Step 2: Wait for connection requests arrival. If there is a connection request $R(a,b)$ arrival, go to step 3; otherwise, update the network state and go back to step 2.

Step 3: Compute a virtual route a'–x'–y'–b'(x' and y' denote some virtual nodes). If this route can be found successfully, go to step 4; otherwise, block this request, update the network state and go back to step 2.

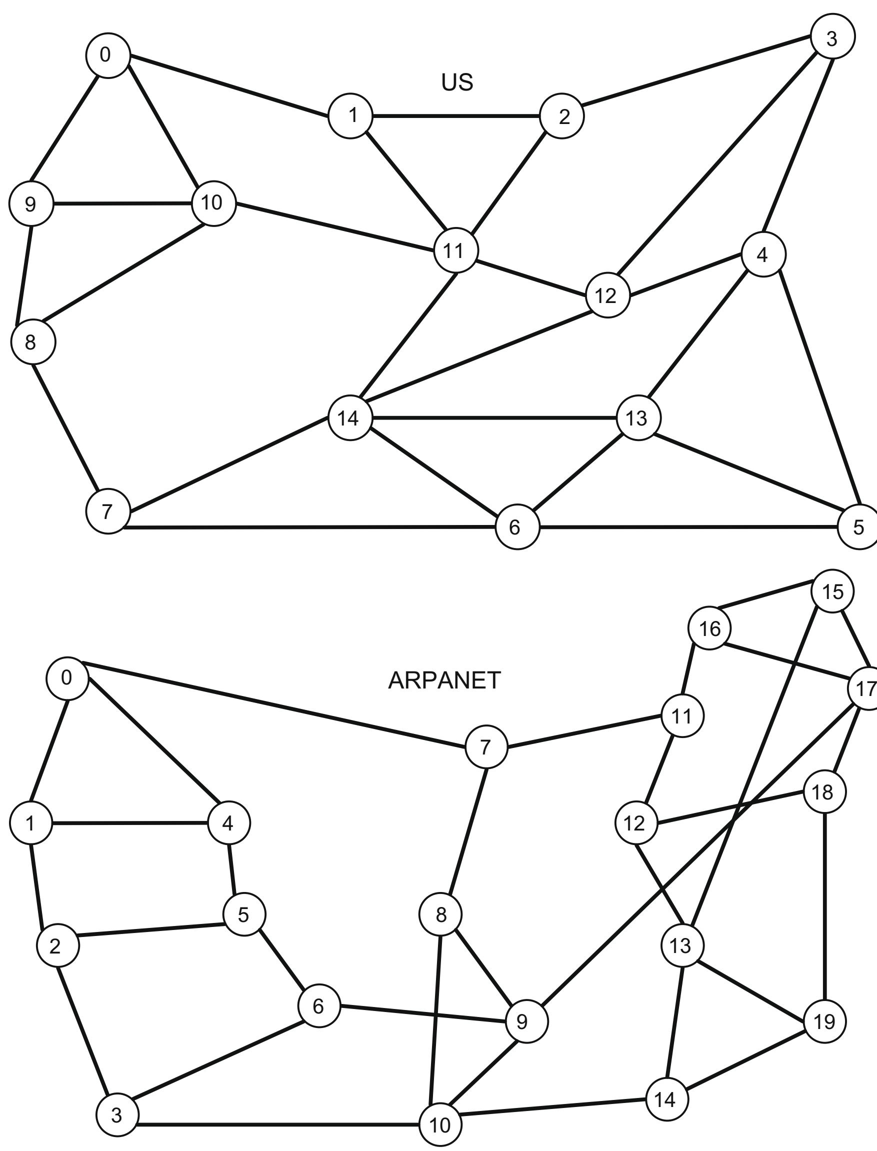

Fig. 2. Network topologies in simulation.

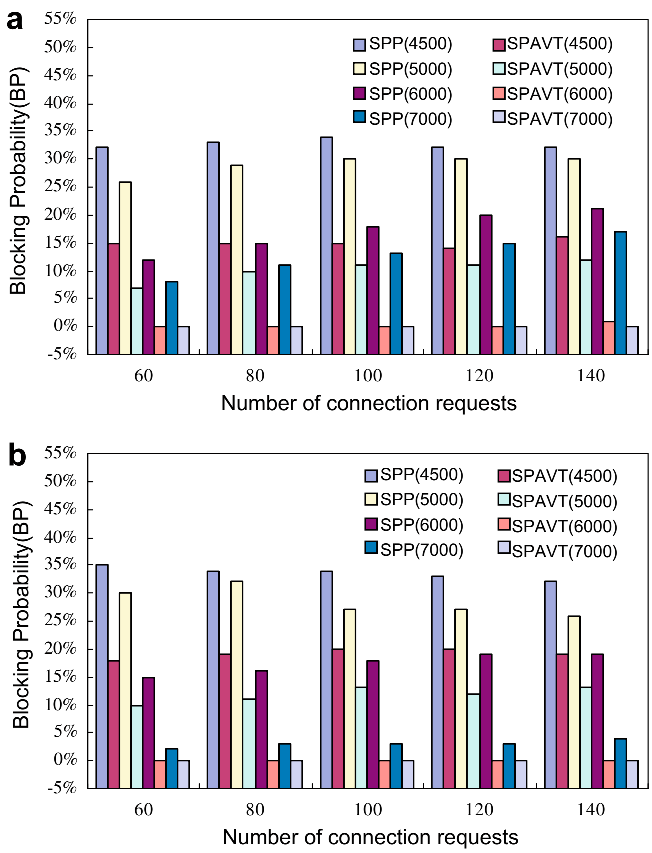

Fig. 3. Performance comparison in blocking probability of SPP and SPAVT in (a) US and (b) ARPANET networks.

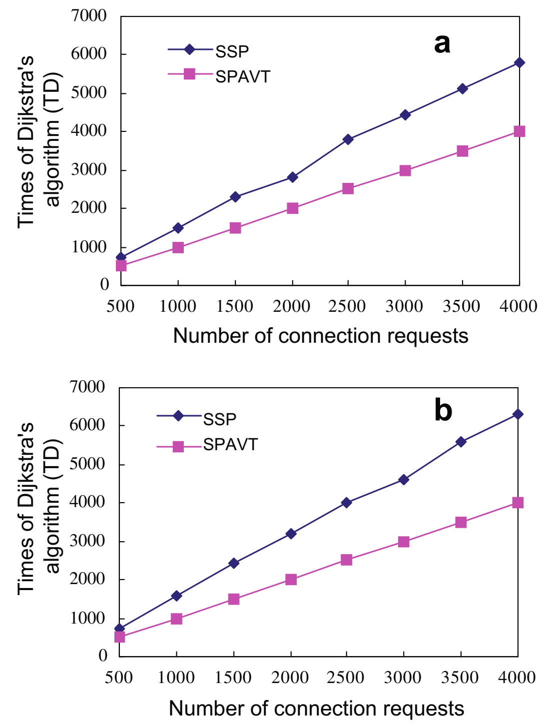

Fig. 4. Performance comparison in time complexity of SPP and SPAVT in (a) US and (b) ARPANET networks.

Step 4: With the consideration of consuming the least working resources, choose the sub-primary paths $K_p^{a',x'}[i](1 \leqslant i \leqslant |K_p^{a',x'}|)$, $K_p^{x',y'}[j](1 \leqslant j \leqslant |K_p^{x',y'}|)$ and $K_p^{y',b'}[s](1 \leqslant s \leqslant |K_p^{y',b'}|)$ in $K_p^{a',x'}$, $K_p^{x',y'}$ and $K_p^{y',b'}$ corresponding to the virtual links (a',x'), (x',y') and (y',b') on the virtual route a'–x'–y'–b'. Then, With the consideration of consuming the least backup resources, based on the shared resources policy, choose the sub-backup paths $K_{b,i}^{a',x'}[m](1 \leqslant m \leqslant |K_{b,i}^{a',x'}|)$, $K_{b,j}^{x',y'}[n](1 \leqslant n \leqslant |K_{b,j}^{x',y'}|)$ and $K_{b,s}^{y',b'}[u](1 \leqslant u \leqslant |K_{b,s}^{y',b'}|)$ in $K_{b,i}^{a',x'}$, $K_{b,j}^{x',y'}$ and $K_{b,s}^{y',b'}$ corresponding to $K_p^{a',x'}[i]$, $K_p^{x',y'}[j]$ and $K_p^{y',b'}[s]$. Go to step 5.

Step 5: Accept this connection request, record the primary path, backup path and the consumed resources, update the network state, and go back to step 2.

The off-line time complexity to initialize the virtual topology is mainly dependent on the times of running the k-shortest algorithm whose time complexity is $O(k \cdot |N|^3)$ according to [9]. In step 1, SPAVT will run one time of the k-shortest algorithm to compute k primary paths for each node pair, and will run one time of the k-shortest algorithm to compute k backup paths for each primary path. In the worst case, there will be $|N|^2$ node pairs in the network, and then the off-line time complexity of SPAVT is approximately $O(k^2 \cdot |N|^8)$.

The on-line time complexity of SPAVT is mainly dependent on the times of running the Dijkstra's algorithm whose time complexity is $O(|L| + |L| \cdot \log|N|)$ according to [10]. Since SPAVT will only run one time of the Dijkstra's algorithm to compute the virtual route on virtual topology in step 3 for each connection request, the on-line time complexity of SPAVT is approximately $O(|L| + |L| \cdot \log|N|)$.

4. Simulation results and analysis

In simulation, the test networks are US and ARPANET networks as shown in Fig. 2, where US network has 15 nodes and 27 links and ARPANET networks has 20 nodes and 31 links. We assume each link is bidirectional and has 100 wavelengths, and each network node has the full wavelength conversion capacity. We consider the incremental traffic model. The basic cost of each link is assumed to 1. In this paper, we evaluate two performance parameters: (1) blocking probability (BP) is the ratio of the number of blocked requests over the number of all requests; (2) times of Dijkstra's algorithm (TD) is the number of the times of running the Dijkstra's algorithm for searching paths. We compare the proposed SPAVT with previous SPP and SSP [2–8].

In Fig. 3, the Y in SPP(Y) and SPAVT(Y) indicates the value of the protection-switching time constraint with unit microsecond (μs). We can see from Fig. 3 that, when $Y = 4500$ μs, whether in US or ARPANET networks, the BP of SPP is much higher than that of SPAVT. The reason for this is that: (1) the protection-switching time constraint 4500 μs is small, so that the number of requests blocked by dissatisfying the protection-switching time will increase in SPP; (2) SPAVT is a special case of sub-path protection and it can well satisfy the protection-switching time constraint by searching paths in advance and computing a virtual route in virtual topology. When $Y = 7000$ μs, the BP of SPAVT is close to 0, while that of SPP is still big. It is shown that, if the protection-switching time is set to a suitable value, SPAVT can solve the problem of blocking connection requests caused by dissatisfying the protection-switching time constraint in SPP.

We can see from Fig. 4 that, whether in US or ARPANET networks, comparing with SSP, SPAVT significantly reduce the times of running the Dijkstra's algorithm when routing computation, and the improvement ratio is 30–50%. The reason for this has been presented in Section 2.3. Therefore, we can say that SPAVT has much lower time complexity than SSP.

5. Conclusion

In this paper, we have considered the protection-switching time and algorithm time complexity, and proposed a new survivable algorithm, sub-path protection based on auxiliary virtual topology (SPAVT) to tolerate the single-link failure in WDM optical networks. SPAVT first searches multiple pairs of primary and backup paths for each node pair in the network by the off-line manner according to the protection-switching time constraint, and then map these paths to the virtual topology. When a connection request arrives, SPAVT only needs to run one time of the Dijkstra's algorithm to search a virtual route in virtual topology, where the route may consist of multiple pairs of sub-paths, to meet the protection-switching time constraint. Then, according to the shared resources policy, SPAVT chooses an optimal pair of sub-paths. Simulation results show that, compared with conventional algorithms, SPAVT can reduce the blocking probability and perform lower time complexity.

References

[1] D. Zhou, S. Subramaniam, Survivability in optical networks, IEEE Network 14 (2000) 16–23.
[2] S. Ramamurthy, L. Sahasrabuddhe, B. Mukherjee, Survivable WDM mesh networks, Journal of Lightwave Technology 21 (2003) 870–883.
[3] L. Guo, J. Cao, H. Yu, L. Li, Path-based routing provisioning with mixed shared protection in WDM mesh networks, Journal of Lightwave Technology 24 (2006) 1129–1141.
[4] C. Ou, J. Zhang, H. Zang, L. Sahasrabuddhe, B. Mukherjee, New and improved approaches for shared-path protection in WDM mesh networks, Journal of Lightwave Technology 22 (2004) 1223–1232.
[5] P. Ho, T. Hussein, A framework for service-guaranteed shared protection in WDM mesh networks, IEEE Communication Magazine 40 (2002) 97–103.
[6] M. Tornatore, M. Carcagnì, C. Ou, B. Mukherjee, A. Pattavina, Intelligent shared-segment protection, Computer Networks 52 (2008) 1965–1974.
[7] D. Xu, Y. Xiong, C. Qiao, Novel algorithms for shared segment protection, IEEE Journal on Selected Areas in Communications 21 (2003) 1320–1331.
[8] P. Ho, J. Tapolcai, T. Cinkler, Segment shared protection in mesh communications networks with bandwidth guaranteed tunnels, IEEE/ACM Transactions on Networking 12 (2004) 1105–1118.
[9] J. Yen, Finding the k shortest loopless paths in a network, Management Science 17 (1971) 712–716.
[10] M. Barbehenn, A note on the complexity of Dijkstra's algorithm for graphs with weighted vertices, IEEE Transactions on Computers 47 (1998) 263.