Candidate-Cycle-based Heuristic Algorithm for Node-and-Link Protection of Dynamic Multicast Traffic in Optical DWDM Networks

Ahmed Frikha, Samer Lahoud, and Bernard Cousin
University of Rennes 1, IRISA, 35042 Rennes Cedex, France
Email:{afrikha,slahoud,bcousin}@irisa.fr

Abstract—Maintaining survivability of DWDM networks is crucial to multicast traffic. A link-or-node failure has a severe impact on optical multicast sessions as it can prune several communications simultaneously. In this paper, we present a novel candidate-cycle-based heuristic algorithm for node-and-link protection (CCHN) in dynamic multicast traffic. CCHN is based on p-cycle protection concept. The p-cycle concept ensures a fast restoration time and an efficient use of network capacity. Extensive simulations show that the blocking probability of our algorithm is lowest. Furthermore, the computational time of our algorithm is very low compared with the existing approaches, especially when traffic load is high.

I. INTRODUCTION

Optical DWDM network provides a high bandwidth as it allows hundreds of wavelengths to be multiplexed onto a single fiber. Therefore, it is important to maintain DWDM network survivability since a single link-or-node failure would affect a large number of communication sessions. In multicast communications, this impact is more severe as a link-or-node may carry traffic for multiple destinations. Hence, protecting multicast sessions in DWDM networks is a crucial task.

Five major multicast protection approaches are proposed in the literature and most of them focus on link failure recovery: 1) tree-based protection [1]-[2]-[3], 2) path-based protection [4]-[5], 3) segment-based protection [6]-[7], 4) ring-based protection [8]-[9], and 4) p-cycle (preconfigured protection cycle) based protection [10]-[11]. In [12], G. Xue et al. propose to deploy two link-disjoint light-trees: an active light-tree and a backup light-tree. However, identifying two link-disjoint light-trees is not always possible in networks where the average nodal degree is low. Moreover, this approach is not efficient in term of network capacity utilization as backup light-tree sharing is not always possible. Work in [1] relaxes the disjointness constraint from link-disjoint to arc-disjoint, but this proposition suffers from the same problems. In 2003, N. Singhal et al. proposed two algorithms based on the optimal path pair protection approach (OPP): 1) the OPP-based shared disjoint path (OPP-SDP) algorithm [8], 2) the OPP-based shared disjoint segment (OPP-SDS) algorithm [8]. Note that the OPP approach is based on the algorithm described in [13] to compute a pair of link-disjoint paths from a source to every destination. The OPP-SDP algorithm implements the path-based approach and allows a path-pair (backup path and primary path) to share links with already found path pairs. The OPP-SDS algorithm implements the segment-based approach and allows backup segments to share links with other backup and primary segments. The OPP-SDP algorithm was reported to be the most efficient protection algorithm for dynamic multicast traffic. Both of path-based protection and segment-based protection approaches allow more efficient resource utilization. However, these approaches suffer from the low signaling problem, which affects the restoration time. In fact, link failure must be signaled to the extremity of the backup path/segment to handle the restoration process. The ring-based approach provides a fast restoration time, but the resource utilization is not efficient. The p-cycle concept introduced by W.D. Grover [14] for unicast traffic ensures a fast restoration time since p-cycles are pre-cross-connected. When a link fails, the restoration process is handled by the end nodes of the failed link. Moreover, p-cycle protection approach provides a high capacity efficiency as it allows both on-cycle and straddling link to be protected by the cycle. In 2007, F. Zhang and W.D. Zhong showed in [11] that applying p-cycle protection concept for multicast traffic leads to the lowest blocking probability among the aforementioned approaches.

Up to now, most of existing researches in optical multicast traffic focus on link failure recovery and rarely on node failure recovery. Although node failures are less frequent than link failures, node failures may cause the disruption of multiple communications, especially when the failed node is a splitting node for multicast sessions. In 2009, F. Zhang and W.D. Zhong proposed a heuristic algorithm for combined node-and-link failure recovery, named ESHT [15]. The ESHT algorithm is based on p-cycle concept. Then, in [16], they proposed an enhanced version of ESHT, named ESNH, for combined node-and-link failure recovery. The ESHN algorithm achieves the best resource utilization and has the lowest blocking probability among the OPP-SDP algorithm and the ESHT algorithm in a dynamic multicast traffic. However, the computational time of this algorithm is not suitable with a dynamic multicast traffic.

In this paper, we consider the link-and-node failure recovery in a dynamic multicast traffic. We study the ESHN algorithm; we identify the weakness point of this algorithm that affects its computational time in a high traffic load. Then, we propose a candidate-cycle-based heuristic algorithm for link-and-node protection (CCHN). CCHN overcomes the limitations

of ESHN, first by reducing the computational time, and second by enhancing the resource utilization, basing on well selected sets of candidate cycles. We propose two approaches for selecting candidate cycles for the CCHN algorithm. The first approach, named used-cycle-set-based approach (UC-CCHN), attempts to protect the traffic requests by reusing the previously established p-cycles, while the second approach, named l-highest-PC-cycle-set-based approach (PC-CCHN), selects the candidate cycles in advance using a protection capacity score (PC). Simulation results show that both of the proposed approaches UC-CCHN and PC-CCHN outperform the ESHN algorithm in terms of computational time and blocking probability, especially when the traffic load is high.

The rest of this paper is organized as follows. In section II, we discuss the ESHN algorithm for combined node-and-link failure recovery. In section III, we present our novel protection algorithm CCHN as well as the two proposed approaches UC-CCHN and PC-CCHN for selecting candidate cycles. Performed simulations and numerical results are presented in section IV, and we conclude the paper in section V.

II. THE EFFICIENCY SCORE BASED HEURISTIC ALGORITHM OF NODE-AND-LINK PROTECTING P-CYCLE DESIGN (ESHN)

A. Concept

The ESHN algorithm is developed for both static and dynamic multicast traffic protection. In this paper, we consider the ESHN algorithm applied to a dynamic multicast traffic. Work in [16] presents two approaches of ESHN: the first approach, named ESHN-LR, considers only the link failure recovery, while the second approach, named ESHN-CNLR, considers the link-and-node failure recovery. In this paper, we focus on the ESHN algorithm for combined link-and-node failure recovery (ESHN-CNLR). In the following, we note it ESHN, simply.

The ESHN algorithm adapts the efficiency-ratio based unity-p-cycle heuristic algorithm (ERH) [17] to deal with node-and-link failures in multicast traffic. It introduces a new efficiency-score (ES) that measures the efficiency of the cycles in the network. This score takes in consideration the largest amount of unprotected node transit capacity as well as the largest amount of unprotected working link capacity of the multicast tree that a unity-p-cycle can protect. A unity-p-cycle is a p-cycle in the network that reserves only one bandwidth unity (e.g. one wavelength) on each traversed link. Let us consider a multicast light-tree T to protect. Let L denote the unprotected working link capacity of T, N denote the unprotected intermediate node transit capacity of T, and C_j be a unity-p-cycle in the network. The score ES of C_j is given by equation (1), where $W_{j,L}$ is the largest amount of unprotected link capacity in L that C_j can protect, $W_{j,N}$ is the largest amount of unprotected node transit capacity in N that C_j can protect, and $|C_j|$ is the spare capacity required for setting up a unity-p-cycle C_j. $|C_j|$ is given by the number of

links traversed by C_j.

$$ES(C_j) = \frac{W_{j,L} + W_{j,N}}{|C_j|} \qquad (1)$$

After routing a multicast request, ESHN identifies the amount of unprotected working link capacity L and the amount of unprotected node transit capacity N of the computed light-tree. Then, ESHN uses the ES-based unity-p-cycle procedure to protect L and N. The ES-based unity-p-cycle procedure calculates the score ES of each unity-p-cycle and selects the cycle with maximum ES. The amount of working link capacity protected by the selected cycle is subtracted from L and the amount of protected node transit capacity is subtracted from N. This process is iterated until the amount of working link capacity in L and the amount of node transit capacity in N are protected, i.e. $L = \phi$ and $N = \phi$. The selected unity-p-cycles are configured and the corresponding wavelengths are reserved. Note that the reserved cycles (established p-cycles) may serve to protect next coming multicast requests. The flow chart of ESHN is presented in Fig. 1. The reserved capacity of an established p-cycle in the network is released when the p-cycle does not protect any working link capacity and any node transit capacity in the network.
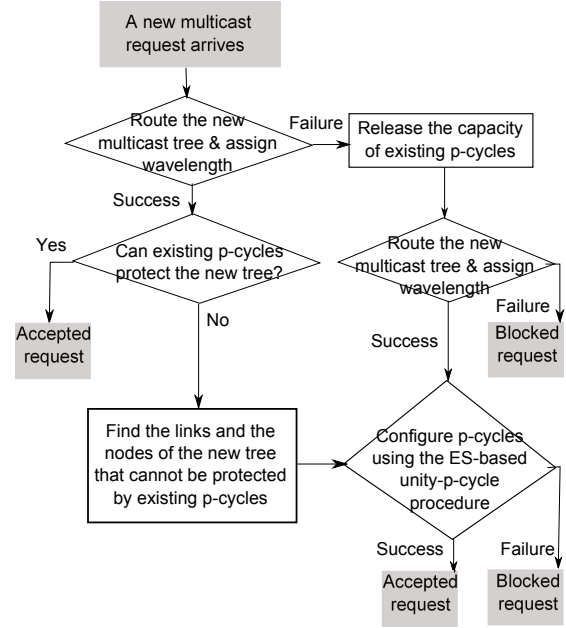

Fig. 1. Flow chart of the ESHN algorithm for combined node-and-link failure recovery in a dynamic multicast traffic

B. Discussion

As described in Fig. 1, when ESHN fails to route the new multicast request, it attempts to perform a global optimization. First, it releases the capacity of existing p-cycles. Then, it routes the new multicast tree. Finally, this algorithm attempts to protect all multicast trees: the working link capacity as well as the node transit capacity of all multicast trees. Note that, for the ESHN algorithm, a cycle can protect an intermediate

node of a tree, if it traverses one of its upstream nodes and all its one-level-downstream nodes in the tree. Moreover, the backup path in the cycle must not traverse the protected node. The complexity of the global optimization procedure is determined by the complexity of protecting all multicast trees simultaneously. At each iteration of the ES-based unity-p-cycle procedure, ESHN has to select the cycle with maximum ES. Therefore, ESHN must compute the ES of each cycle at each iteration. To compute the ES of a cycle, ESHN verifies for each intermediate node of each multicast tree, if the cycle traverses one of its upstream nodes and all its one-level-downstream nodes. Let us compute the complexity of just one iteration of the ES-based unity-p-cycle procedure. Let C be the total number of cycles in the network, T_{unpro} be the number of unprotected multicast trees in the network, N_{unpro} be the maximum number of unprotected intermediate nodes in a multicast tree, $N_{upstream}$ be the maximum number of upstream nodes of an intermediate node in a multicast tree, $N_{downstream}$ be the maximum number of one-level-downstream nodes of an intermediate node in a multicast tree, and L_{unpro} be the total number of unprotected links in all multicast trees. The worst case complexity of computing the ES score of each unity-p-cycle is calculated in 2.

$$C(T_{unpro}N_{unpro}(N_{upstream} + N_{downstream}) + L_{unpro}) \quad (2)$$

This complexity is very high due to the great number of cycles in a network, especially in large networks or when the average nodal degree of the network is high. Therefore, protecting all multicast trees simultaneously to perform a global optimization using this method is not feasible in a reasonable computational time.

III. THE CANDIDATE-CYCLE-BASED HEURISTIC ALGORITHM FOR NODE-AND-LINK PROTECTION (CCHN)

A. Concept

In the previous section, we showed that protecting all multicast trees simultaneously using the method described in the flow chart of ESHN is not feasible in a reasonable time. To speed up the computational, we design a novel candidate-cycle-based heuristic algorithm for node-and-link protection (CCHN). Fig. 2 details the operations performed by the CCHN algorithm. As shown in the flow chart, when CCHN fails to route a new multicast request, it performs a global optimization: it releases the established p-cycles in the network. Then, it attempts to route the new request; if the routing fails, the request is blocked; otherwise the algorithm tries to protect all the multicast trees using a set of candidate cycles. The candidate cycle set is a set of carefully selected cycles among the total cycle set of the network. The CCHN algorithm applies the ES-based unity-p-cycle procedure on the candidate cycles instead of applying it on the total cycle set. At each iteration of the ES-based unity-p-cycle procedure, the algorithm selects the cycle with maximum ES among the candidate cycle set. This will reduce considerably the computational time of this procedure. Note that the candidate cycles are used only in the global optimization procedure (when routing and protecting

all the multicast requests simultaneously). In the other part of the algorithm (i.e., when routing and protecting only one multicast request), we consider all the cycles of the network. In the following, we propose two approaches for enumerating the candidate cycles.
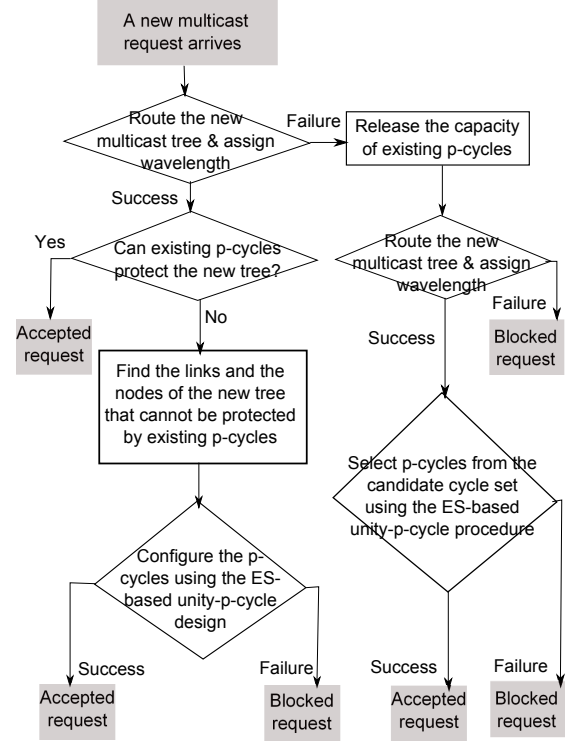

Fig. 2. Flow chart of the CCHN algorithm for combined link-and-node failure recovery in a dynamic multicast traffic

1) The used-cycle-set-based approach (UC-CCHN): The capacity of an existing unity-p-cycle is released when it does not protect any working node transit capacity and any working link capacity, i.e. all multicast requests that this cycle protects have left the network (its holding time is over). Note that this released cycle was previously selected using the ES-based unity-p-cycle procedure to protect a previous request. Therefore, this cycle could be efficient to protect next coming multicast trees as it had the maximum ES. The set of released cycles, named the used cycle set, are stored in a database. The used-cycle-set-based approach (UC-CCHN) consists in using the stored cycles as a candidate cycle set to protect all multicast trees whenever the CCHN algorithm needs to protect all multicast requests simultaneously. Note that at the beginning the used-cycle-set is empty. This does not cause any problem as at the beginning there is no established request and we do not need to execute a global optimization.

2) The l-highest-PC-candidate-cycle-set-based approach (PC-CCHN): To select a candidate cycle set, we define a new score, named protection capacity PC, for each cycle in the network. This score is computed in advance for each cycle before routing the requests. The score PC of a unity-p-cycle C_j, specified by equation (3), is defined as the ratio of the

largest amount of link capacity on the network LC_j that C_j can protect over the sum of spare capacity required by C_j.

$$PC(C_j) = \frac{LC_j}{|C_j|} \qquad (3)$$

A cycle with a high PC, is useful as it maximizes the amount of protected capacity while reserving less spare capacity. The l cycles with highest PC are selected as candidate cycle set, where l is a parameter for the algorithm. The goal of selecting this set is to maximize the capacity that can be protected on the network, and this will help to protect the next coming requests. The l-highest-PC-candidate-cycle-set-based approach (PC-CCHN) consists in using the l selected cycles as a candidate cycle set.

B. Discussion

When the CCHN algorithm fails to route the new multicast request, it attempts to perform a global optimization. It releases all existing p-cycles in the network, routes the new request, and tries to protect all existing multicast trees using a candidate cycle set. As in the ESHN algorithm, the complexity of the global optimization procedure is determined by the complexity of protecting all multicast trees simultaneously. However, unlike the ESHN algorithm, the global optimization procedure of the CCHN algorithm is rapid thanks to the candidate cycles. In fact, applying the ES-based unity-p-cycle procedure on a candidate cycle set will decrease the complexity of this operation. The complexity of one iteration of the ES-based unity-p-cycle procedure is specified by equation (4), where C_c denotes the number of candidate cycles.

$$C_c(T_{unpro}N_{unpro}(N_{upstream} + N_{downstream}) + L_{unpro}) \quad (4)$$

The number of candidate cycles C_c is very low compared to the total number of cycles C, e.g in the European network COST-239 $C = 5058$ (2029 in each direction), while $C_c \prec 200$ in the worst case of our simulations. Thus, the complexity of our proposed algorithm CCHN is very low compared to that of the ESHN algorithm. Consequently, protecting all multicast trees using a candidate cycle set is feasible in a reasonable computational time.

IV. PERFORMANCE EVALUATION

In this section, we evaluate our proposed algorithm CCHN for combined link-and-node failure recovery by comparison with the ESHN algorithm. We assess performance of the two proposed approaches of CCHN: the PC-CCHN approach and the UC-CCHN approach. In our simulation, we assume that the global network traffic is a combination of unicast and multicast traffic. Multicast traffic requests are generated with a probability R, and unicast traffic requests are generated with a probability $1-R$. The request arrival follows a Poisson process with an average arrival rate λ, and the request holding time follows an exponential distribution with an average holding time μ. Hence, the network offered traffic load is given by $\lambda\mu$. The multicast traffic load is given by $R\lambda\mu$, while the unicast traffic load equals $(1 - R)\lambda\mu$.

We run simulations on the following well known and very often used European optical topologies developed within the COST-266 [18] and COST-239 [19] projects:

- The COST-266 core topology [18] contains 16 nodes and 23 links, with an average nodal degree equals to 2.88. The total number of cycles in this topology equals 236 (118 cycles in each direction).
- The COST-239 topology [19] contains 11 nodes and 26 links, with an average nodal degree equals to 4.727. The total number of cycles in this topology equals 5058 (2029 cycles in each direction).

In our study, without lack of generality we assume that each link has two fibers. The two fibers transmit in opposite directions; 16 wavelengths are available on each fiber. The source and the destinations of each multicast session are randomly selected (uniform distribution law). We choose the number of destinations in each multicast request $D = 5$, which seems to be reasonable as the total number of nodes in the used topologies is lower than 16 nodes. We compare the performance of the algorithms according to the following performance criteria:

- The Blocking Probability (BP) represents the percentage of requests that cannot be routed or protected among the total number of requests.
- The Resource Utilization (RU) represents the percentage of reserved wavelengths in the network among the total number of wavelengths. $RU = \frac{W_R}{E*W}$, where W_R represents the total number of wavelengths reserved in the network, E represents the number of fiber in the network, and W the number of wavelengths per fiber.
- The average Computational Time (CT) required for routing and protecting a traffic request.

Performance criteria BP, RU, and CT are computed according to the traffic load. For each traffic load value, 5×10^5 requests are generated. This number of requests is enough to measure BP, RU, and CT, with a 95% confidence interval. The probability that a traffic request is multicast request $R = 0.23$. This probability is the same used in [16] for evaluating the ESHN algorithm. We believe that this probability is reasonable since the capacity reserved by a multicast traffic request is high compared to the one reserved by a unicast traffic request.

First, we consider the COST-266 topology. The total number of cycles in this topology $C = 236$. This number is low compared to that of the COST-239 topology. Therefore, executing the ESHN algorithm on the COST-266 topology is not a hard task. We choose the number of candidate cycles $l = 100$ for the l-highest-PC-candidate-cycle-set-based approach (PC-CCHN). Fig. 3 illustrates the blocking probability measured in the COST-266 topology. The PC-CCHN approach has a blocking probability slightly lower than that of the ESHN algorithm and that of the UC-CCHN approach. The blocking probability of the UC-CCHN approach is very close to that of the ESHN algorithm. In the UC-CCHN approach, the average number of candidate cycles is lower than l and C,
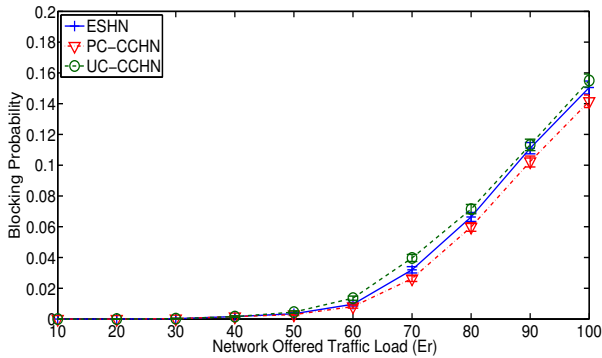
Fig. 3. Blocking probability comparison in COST-266 network
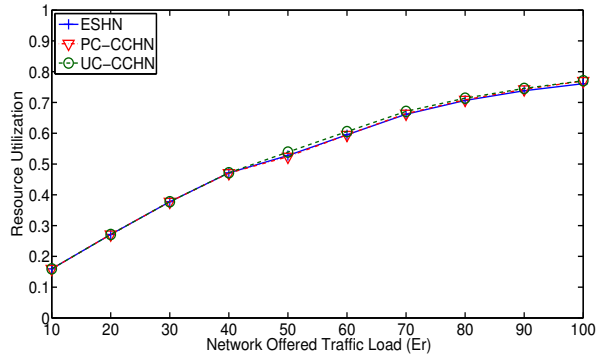

Fig. 4. Resource utilization comparison in COST-266 network

e.g. for a traffic load equals to 100 Erlang, the average number of candidate cycles equals $C_c = 72.7 \pm 3.44$ cycles for the UC-CCHN approach. This low number of candidate cycles makes the blocking probability of the UC-CCHN approach slightly higher than that of ESHN, but also makes its average computational time CT the lowest.

Fig. 4 shows the resource utilization of the algorithms. The percentage of wavelength reserved by the algorithms is almost the same. When the traffic load increases, the percentage of wavelengths reserved per link is higher. For a traffic load equals 100 Erlang, almost 80% of the wavelengths on each link are reserved.

Now, let us focus on the average computational time CT for setting up a traffic request. Table I illustrates the average computational time CT for each algorithm for different values of network offered traffic load. As expected, the UC-CCHN approach is the most rapid among the ESHN algorithm and the PC-CCHN approach. This is due to the low number of candidate cycles. Also, the PC-CCHN approach has a low average computational time compared to ESHN. As shown in Table I, the computational time CT of the ESHN algorithm increases rapidly with the offered traffic load augmentation. In fact, the probability of executing the global optimization procedure, which has a high computational complexity, is higher when the traffic load increases. The PC-CCHN approach outperforms the ESHN algorithm in terms of computational

time and blocking probability, while the UC-CCHN algorithm has a blocking probability close to that of ESHN, and has computational time very low compared to that of ESHN, especially when the traffic load is high.

Traffic Load	60 Er	70 Er	80 Er	90 Er	100 Er
UC-CCHN	12.6 ms	18.1 ms	22.4 ms	26.5 ms	30.7 ms
PC-CCHN	12.7 ms	22.3 ms	36.7 ms	54.3 ms	65.5 ms
ESHN	21.9 ms	34.8 ms	62.5 ms	93.8 ms	116.1 ms

TABLE I
COMPARISON OF THE AVERAGE COMPUTATIONAL TIME CT FOR SETTING UP A REQUEST IN COST-266 NETWORK

To evaluate performance of the algorithms in a more difficult case, we consider the COST-239 topology, which contains 5058 cycles. We choose the number of candidate cycles $l = 200$ for the l-highest-PC-candidate-cycle-set-based approach. We choose the value of this parameter very low compared to the total number of cycles in the network to maintain a reasonable average computational time. The average computational time of the ESHN algorithm is expected to be very high due to the great number of cycles in the network. Therefore, executing ESHN in this topology is not practical. To perform simulations easily, we define a limit T_l for the computational time, $T_l = 100$ seconds. If the computational time for setting up a request exceeds T_l, we stop the computation and the request is blocked.

Fig. 5 illustrates the blocking probability of the algorithms measured in the COST-239 topology according to the network offered traffic load. As shown in this figure, when the traffic load is high, the ESHN algorithm has a blocking probability very high compared to our two proposed approaches PC-CCHN and UC-CCHN. This high blocking probability is due to the great number of requests that are blocked because their computational time is higher than T_l. The blocking probability of the PC-CCHN approach is very close to that of the UC-CCHN approach and is lower than 4×10^{-2} for a network traffic load equal to 200 Erlang, while the blocking probability of the ESHN algorithm is higher that 15×10^{-2}. The efficiency of our algorithms PC-CCHN and UC-CCHN in terms of resource utilization is proven in Fig. 6. This figure shows that the RU of the PC-CCHN and UC-CCHN algorithms is lower than that of the ESHN algorithm. The percentage of wavelengths reserved on each link by the PC-CCHN approach is the lowest. In fact, the p-cycles reserved using the l-highest-PC-candidate-cycle-set-based approach provide more protecting capacity in the network while reserving less spare capacity. This will reduce the resource utilization as well as the blocking probability of this algorithm.

To assess the rapidity of our proposed approaches, we measure the average computational time CT for setting up a traffic request. Note that we did not measure the average CT for the ESHN algorithm as we stop the computation when it is longer than $T_l = 100$ seconds. In fact, the computational time of the ESHN algorithm is always higher than 100 seconds when the algorithm try to perform a global optimization. As
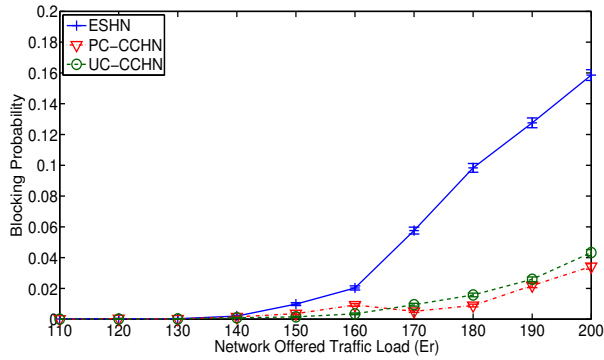
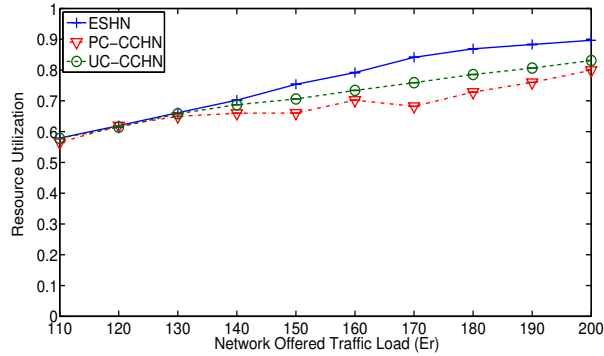Fig. 5. Blocking probability comparison in COST-239 network

Fig. 6. Resource utilization comparison in COST-239 network

shown in Table II, the PC-CCHN approach is slightly faster than the UC-CCHN approach when the traffic load is higher than 180 Erlang. Both of the proposed approaches have a low computational time, and this proves that performing the global optimization is practical using a candidate cycle set.

Traffic Load	160 Er	170 Er	180 Er	190 Er	200 Er
UC-CCHN	38.3 ms	36.8 ms	35 ms	34.8 ms	34.8 ms
PC-CCHN	38.6 ms	39.6 ms	32.9 ms	29.6 ms	25 ms

TABLE II
COMPARISON OF THE AVERAGE COMPUTATIONAL TIME CT FOR SETTING UP A REQUEST IN COST-239 NETWORK

V. CONCLUSION

In this paper, we investigated the link-and-node failure recovery for a dynamic multicast traffic in optical DWDM networks. We analyzed the p-cycle based protection algorithm ESHN. The ESHN algorithm was reported to be the most efficient algorithm for node-and-link failure recovery in a dynamic multicast traffic. Analytical results prove that executing the ESHN algorithm is not practical when the traffic load is high, especially when the total number of cycles in the network is high. In such a case, the computational time of ESHN for setting up a traffic request is very long. We proposed a novel algorithm, named CCHN, for combined node-and-link failure recovery. This algorithm speeds up the computational time of setting up a traffic request by enumerating a set of candidate

cycles. Precisely, we proposed two approaches for selecting the candidate cycles. The UC-CCHN approach attempts to protect traffic requests by reusing the previously established p-cycles, while the PC-CCHN selects the candidate cycles in advance based on the protection capacity score PC. Both of the proposed approaches provide a blocking probability very low compared to that of ESHN, especially in a very high traffic load. Moreover, our approaches are faster than the ESHN algorithm.

REFERENCES

[1] N. K. Singhal, C. Ou, and B. Mukherjee, Cross-sharing vs. self-sharing trees for protecting multicast sessions in mesh networks, *in proceedings Comput. Netw.*, vol. 50, no. 2, pp. 200-206, 2006.
[2] M. Y. Saidi, B. Cousin, M. Molnar, Improved Dual-Forest for Multicast Protection, *in proceedings 2nd Conference on Next Generation Internet Design and Engineering Conference, Valencia, Spain,*, pp. 371-378, Apr. 2006.
[3] M. Medard, S. G. Finn, R. A. Barry, and R. G. Gallager, Redundant trees for preplanned recovery in arbitrary vertex-redundant or edge-redundant graphs, *IEEE/ACM Trans, Netw.*, vol. 7, no. 5, pp. 641-652, Oct. 1999.
[4] T. Rahman and G. Ellinas, Protection of multicast sessions in WDM mesh optical networks, *in proceedings OFC'05, Anaheim, CA,* p. 3, Mar. 2005.
[5] P. Leelarusmee, C. Boworntummarat, and L. Wuttisittikulkij, Design and analysis of five protection schemes for preplanned recovery in multicast WDM networks, *in proceedings IEEE SAWWC' 04, Princeton, NJ,* pp. 167-170, Apr. 2004.
[6] H. B. Luo, H. F. Yu, L. M. Li, and S. Wang, On protecting dynamic multicast sessions in survivable mesh WDM networks, *in proceedings OFC'06, Anaheim, CA,* p. 3, Mar. 2006.
[7] N. K. Singhal and B. Mukhejee, Dynamic provisioning of survivable multicast sessions in optical WDM mesh networks, *in proceedings Optical Fiber Communications Conference,* pp. 207-209, Mar. 2003.
[8] N. K. Singhal, L. H. Sahasrabuddhe, and B. Mukherjee, Provisioning of survivable multicast sessions against single link failures in optical WDM mesh networks, *J. Lightw. Technol.,* vol. 21, no. 11, pp. 2587-2594, Nov. 2003.
[9] C. Lu, H. Luo, S. Wang, and L. M. Li, A novel shared segment protection algorithm for multicast sessions in mesh WDM networks, *ETRI J.,* vol. 28, pp. 329-336, 2006.
[10] W. D. Zhong, F. Zhang, and Y. H. Jin, Optimized designs of p-cycles for survivable multicast sessions in optical WDM networks, *in proceedings ChinaCom'07, Shanghai, China,* Aug. 2007.
[11] F. Zhang and W. D. Zhong, Applying p-cycles in dynamic provisioning of survivable multicast sessions in optical WDM networks, *in proceedings Optical Fiber Communications Conference, Anaheim, CA,* Mar. 2007.
[12] G. Xue, L. Chen, and K. Thulasiraman, Quality-of-service and quality-of-protection issues in preplanned recovery schemes using redundant trees, *IEEE J. Sel. Areas. Commun.,* vol. 21, no. 8, pp. 1332-1345, 2003.
[13] J.W. Suurballe. Disjoint paths in a network. *Networks,* vol. 4, pp. 125-145,1974.
[14] W. D. Grover and D. Stamatelakis, Cycle-oriented distributed preconfiguration: ring-like speed with mesh-like capacity for self-planning network restoration, *in proceedings of IEEE ICC,* 1998.
[15] F. Zhang, and W. D. Zhong, p-Cycle based tree protection of optical multicast traffic for combined link and node failure recovery in WDM mesh networks, *IEEE Commun. Lett.,* vol. 13, no. 1, pp. 40-42, Jan 2009.
[16] F. Zhang and W. D. Zhong, Performance evaluation of optical multicast protection approaches for combined node and link failure recovery, *J. Lightw Technol.,* vol. 27, no. 18, pp. 4017-4025, 2009.
[17] Z. R. Zhang,W. D. Zhong, and B. Mukherjee, A heuristic method for design of survivable WDM networks with p-cycles, *IEEE Commun. Lett.,* vol. 8, pp. 467-469, 2004.
[18] S. De Maesschalck et al., Pan-European Optical Transport Networks: an Availability based Comparison, *Photonic Network Communications,* Vol. 5, no. 3, pp. 203-226, May 2003.
[19] P. Batchelor et al.: Ultra High Capacity Optical Transmission Networks. *Final report of Action COST 239,* 1999.