2009 First International Conference on Future Information Networks

Local-map-based Candidate Node-Encircling Pre-configuration Cycles Construction in Survivable Mesh Networks

Taifei Zhao, Xizheng Ke, Peilin Yang
Faculty of Automation and Information Engineering Xi'an University of Technology, 710048
Xi'an, China
zhaotaifei@163.com

***Abstract*—This paper studies the protection problem of pre-configuration cycles (p-cycles) in survivable mesh networks. A new algorithm called Local-map-based Finding p-Cycles Algorithm (LFCA) is proposed to find the candidate p-cycles. The major difference between the previous algorithms of finding cycles and LFCA is that LFCA can find node-encircling p-cycles and some special link p-cycles which must contain some fixed nodes. The performance of LFCA is evaluated by computer simulations on the real world network topology. ;**

Keywords- mesh network; p-cycle; local-map; finding cycles

I. INTRODUCTION

WDM (Wavelength Division Multiplexing) technology can multiplex hundreds of wavelengths onto a single fiber for concurrent data transmission and the capacity of a WDM wavelength can easily reach 10Gb/s. So, WDM optical networks will provide the backbone infrastructure for future communications. Due to the high-speed nature of optical networks, any accidental failure such as a fiber-cut will result in gigantic data loss. Therefore, it is imperative that the network can survive from the failure and achieve fast optical recovery.

The concept of Pre-configuration Cycle (p-cycle) is first proposed by Grover and Stamatelakis in 1998. P-cycle is a promising approach for protecting working capacities in WDM networks because of its ability to achieve ring-like recovery speed while maintaining the capacity efficiency of a mesh-restorable network [1]. A p-cycle is a cyclic pre-connected closed path of spare capacity which provides protection for any span whose end nodes are both on the cycle. A node-encircling p-cycle has to include all adjacent nodes of a central node, so it can also recover from the central node failures and straddling flow failures [2]. Thus, it is an important problem to find a set of candidate p-cycles to protect a given working capacity distribution. Then, we can configure the p-cycles in the mesh networks. Node-encircling p-cycles are routed through all adjacent neighbor nodes of the so-called central node which is considered as a prospective failure node [3]. Then the node-encircling p-cycles can protect against the failure of the central node by providing an alternate path among all of the adjacent neighbor nodes. So, finding the candidate node-encircling p-cycles is the fist step of the node-encircling p-cycles design in mesh networks.

This research is supported by China Postdoctoral Science Foundation (Grant No.20080441179) and Scientific research plan projects of Shaanxi Education Department (Grant No. 07JK332, No. 08JK386)

978-1-4244-5160-9/09/$26.00

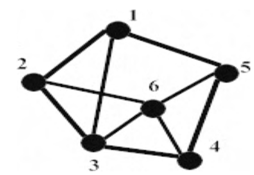

Figure 1. Node-encircling p-cycle

There are two basic types of p-cycle, the link p-cycle is able to protect the working capacity of a link and the node-encircling p-cycle can protect all traversing paths through a single node. Fig. 1 shows an example of a network with the p-cycles. The p-cycle 1-2-3-6-5-1 is a link p-cycle which can protect five on-cycle spans (1-2, 2-3, 3-6, 6-5, 5-1) and two straddling spans (1-3, 2-6). The p-cycle 1-2-3-4-5-1 is a simple node-encircling p-cycle which is able to protect five on-cycle spans (1-2, 2-3, 3-4, 4-5, 5-1) and one straddling span (1-3). This node-encircling p-cycle can also provide protections for the central node 6 and straddling flow such as 2-6-4, 3-6-5, 4-6-5 and etc.

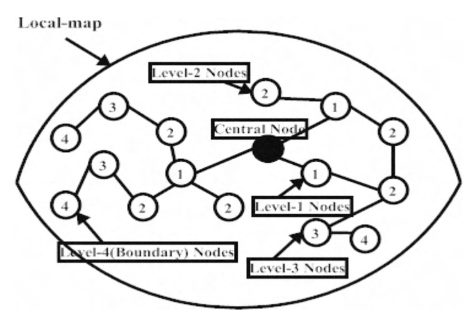

Figure 2. A sample level-4 local-map

The concept of local-map is proposed in [4]. In the network, the central nodes can be designated by the administrator and each central node keeps a small-sized data table known as the local-map. The data table stores network information such as the node connectivity and the cost/distance of available links. A local-map of the central node is built before finding the candidate p-cycles and updated after configuring the p-cycles. The size or scope of a local-map is measured by the maximum number of hops or the maximum distance from the central node to the other nodes. According to the two measure metrics, the local-map can be denoted as "level local-map" and "max-cost local-map" respectively throughout this paper. An example of level-4 local-map with a certain central node is illustrated in Fig. 2. By studying some existing telecom networks, we found that a local-map of level 3 or 4 is sufficient to construct the node-encircling p-cycles.

An ILP-based p-Cycle construction algorithm without candidate cycle enumeration is proposed in [5]. This algorithm can only be used when the required number of p-cycles is not too large. Another approach to generate candidate p-cycles is proposed in [6]. This approach finds all fundamental cycles and constructs candidate p-cycles by merging fundamental cycles. This method still based on ILP formulation with associated time complexity. In reference [7], a method use Integer Linear Program (ILP) to find the simple node-encircling p-cycles, which have to use the enumeration of all cycles. The Weighted DFS-based Cycle Search (WDCS) algorithm based on Depth First Search (DFS) algorithm is proposed in [8], but the efficiency of finding cycles is not very good. In this paper, we propose a novel heuristic algorithm called the Local-map-based Finding p-Cycles Algorithm (LFCA), which can be used to find more good candidate simple node-encircling candidate p-cycles of a central node in the mesh networks.

II. CONSTRAINTS AND BASIC OPERATIONS OF LFCA

A. Notations and Assumptions

We define a network topology G(N, E) for a given optical mesh network, where N is the set of nodes, and E is the set of bidirectional edges. |N| and |E| denote the node number and the edge number respectively. A bidirectional edge consists of two directed edges. We use bidirectional pair (u, v) to represent a bidirectional edge between u and v and ordered pair [u, v] to represent a directed edge from u to v. Thus a bidirectional edge (u, v) consists of two directed edges [u, v] and [v, u].

To measure the efficiency of a p-cycle in protecting working capacity, define a pre-selection metric called a priori efficiency (AE) which is shown in Eq.(1)[9]. Given a p-cycle p, AE(p) is defined as the number of total working optical paths which has the potential to be protected divided by the total hops of the p-cycle p, i.e.,

$$AE(P) = \frac{\sum_{i \in S} X_{p,i}}{\sum_{i \in S, X_{p,i}=1} h_i} \qquad (1)$$

where S is the set of spans in the network, hi is the hop of span i, and $X_{p,i}$ is the number of restoration paths that p-cycle p can provide for span i, X_p,i=1 if span i is an on-cycle span, $X_{p,i}$=2 if span i is a straddling span, $X_{p,i}$=0 otherwise. A larger AE means higher efficiency because the p-cycle with a large AE can protect more spans.

B. Basic Operations

If the simple node-encircling p-cycles of a central node exist, we could get one node-encircling candidate p-cycle in a local-map of this central node by the Local-map-based Depth First Search (LDFS) algorithm. The key difference between the LDFS and the DFS is that the order of cycle search is controlled by assigning weights to the directed edges of the local-map in the LDFS. So the cycles containing all neighbor nodes of the central node in the local-map are likely to be found early in the search.

The redundant span of a p-cycle is defined as: if a span is removed from the p-cycle, we can find a new p-cycle which contains all the target nodes and does not involve the nodes which are not on the original p-cycle. In Fig. 3, the node 4 is the central node of the node-encircling p-cycle 1-2-3-5-6 and the target nodes are the neighbor nodes (1,2,3,5). For this p-cycle, the span 1-6 and span 6-5 are the redundant spans. The reason is that the new cycle 1-2-3-5 is a smaller hops node-encircling p-cycle for central node 4.

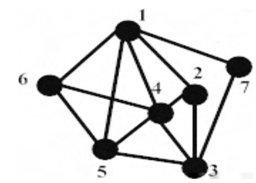

Figure 3. Contraction algorithm

The simple node-encircling p-cycles generated by the LDFS algorithm usually have some redundant on-cycle spans. In order to get the minimal-hops cycles without the redundant spans, we perform the contraction algorithm on these simple node-encircling p-cycles generated by the LDFS. The contraction algorithm begins with an existing simple node-encircling p-cycle. Each on-cycle path of this p-cycle is between the two neighbor nodes of the central node. A new smaller p-cycle is constructed by removing the on-cycle path and replacing it with the shortest hops path between the two end nodes of this path. The shortest hops path should be node-disjoint with the remained part of the p-cycle excluding the on-cycle path. After contracting every on-cycle path, we get the minimal-hops p-cycle which contains all neighbor nodes of the central node. An example of the contraction algorithm is illustrated in Fig. 3 using the node-encircling p-cycle 1-6-4-5-3-7-1 with the central node 2 and the target nodes (1,4,3). The shortest hops path between the target node 4 and the target node 3 is the path 4-3. So we replace the on-cycle path 4-5-3 with the path 4-3 and get the smaller p-cycle 1-6-4-3-7-1. For this cycle, the shortest hops path between the target node 4 and the target node 1 is the path 4-1. Run the contraction algorithm again and get another smaller p-cycle 1-4-3-7. For no redundant span on this p-cycle, the p-cycle 1-4-3-7 is the minimal-hops p-cycle for the central node 2.

In order to get more efficient p-cycles, we can perform the basic SP-Add operation on the minimal-hops cycle. The basic idea of Sp-Add operation is that, for an on-cycle span of an existing cycle, a new cycle is constructed by removing the span and replacing it with the shortest path between its end nodes which is node-disjoint from the original cycle. So the on-cycle span becomes a straddling span of the new cycle being formed. We propose a Local-Extend operation to generate more node encircling p-cycle. The weight of a directed edge [u, v] is denoted as weight[u, v] in the graph. The process of the Local-

Extend operation is as follows: in a local-map, the cost of edge [u, v] is set as -weight[u, v] before a Local-Extend operation. Then we run a least-cost rout algorithm in such local-map to compute a least-cost path between the end nodes of an on-cycle span. The total weight of the edges on this path will be larger. This path is different from the path generated by SP-Add. So we can get a new good candidate p-cycle by replacing the on-cycle span with this more expensive path. It is obviously that the cycles generated by this algorithm contain all the target nodes in the original cycle.

III. LOCAL-MAP-BASED FINDING P-CYCLES ALGORITHM

We prune off those nodes whose nodal degree is less than 2 and all the links adjacent to them in the topological graph, because no simple p-cycle could cover such nodes. The process of the LFCA is presented as follow:

Step 1: Record the neighbor nodes and the level-1 local-map of a given central node. Regard all the neighbor nodes as the target nodes.

Step 2: For every node in the local-map with a queue to record the information of this node, arrange the outgoing arcs of every node in the ascending order of the arcs' cost and push the outgoing arcs whose heads are the target nodes into the top of the corresponding queue. Prune off the given central node and the links adjacent to it in the local-map. Perform the LDFS algorithm in the local-map to generate a cycle C containing all the target nodes.

If such the cycle C can be found and go to Step 3.

Else, this central node does not have any simple node-encircling p-cycle in this local-map.

If the local-map level of this central node can be increased, increase this local-map level and update this local-map, go to Step 2.

Else the local-map level of this central node reaches the upper limit of the local-map level L_{max}. This central node does not have any simple node-encircling p-cycles, give a new central node and go to Step 1.

Step 3: Run Contraction algorithm. Divide the cycle C into the paths with the target nodes which is set as the dividing nodes. Each path created by dividing the cycle C is denoted as the i'th path P_i(i=1,2...K). We compute a shortest hops path P_i between the two end node of the path P_i, P^i should be node-disjoint from all the other paths exclude the path P_i, i.e., $P_j(j \neq i)$. Define the hops difference between the path P^i and the path P_i as hops($P^i - P_i$) and find the maximal value of hops($P^i - P_i$) as hops($P^i - P_i$ $)_{max}$.

If hops($P^i - P_i$ $)_{max} > 0$, we replace the path P_i with the corresponding P_i. Unite all the path P_j(j=1,2...K) to form a new cycle. Let this new cycle as a new cycle C and go to Step 3.

Else, hops($P^i - P_i$ $)_{max} \leqslant 0$, finish running the Contraction algorithm on the cycle C and get the cycle C_{min}, go to Step 4.

Step 4: According to algorithms SP-Expand, SP-Grow and Local-Extend respectively compute the new good candidate cycles based on the cycle C_{min} in the different level local-map. All cycles generated in this step are the simple node-encircling p-cycles with the same central node. According to Eq.(1), compute the AE(p) for each node-encircling p-cycles generated by step 4.

If there are the other central nodes which need to find the simple node-encircling p-cycles, pick up one node as a given central node and go back to Step 1.

Else, record these simple node-encircling p-cycles as the candidate p-cycles.

It is obvious that the simple node-encircling candidate p-cycles generated by the LFCA are the link p-cycles. So this algorithm is also suitable for finding some special link candidate p-cycles which must contain some fixed nodes. When consider the data structure of a spanning tree in a network which has n vertices and n-1 edges. The complexity of a LDFS is O(2n-1). The complexity of the basic span-operation and Local-Extend algorithms is $O(|n|^2)$. In the LFCA, if the cycle C_{min} has m vertices(m n) and the upper limit of the local-map level is L_{max}, the complexity of LFCA for one central node is $O((2n-1) + L_{max}(m\ (n-m)^2))$.

IV. SIMULATION RESULTS AND ANALYSIS

In the study we optimize the Italian network with 21 nodes and 36 links (Fig. 4). Throughout the simulation we use two bi-directional fiber-pairs per link and only consider single link failure on an on-cycle span, a straddling span or straddling flow in this simulation. We assume all the nodes have wavelength (optical–electronic–optical) conversion capacities.

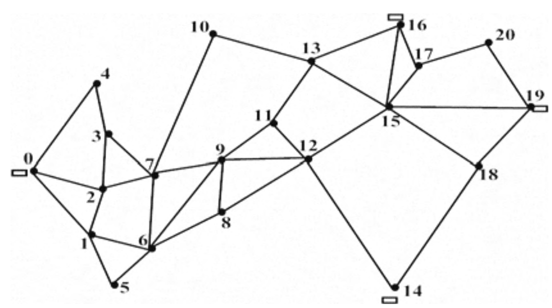

Figure 4. Test topology

In order to finding more candidate p-Cycles, we enhance the local-map level L_{max} as much as possible. So, LFCA is performed in the test topology to find simple node-encircling candidate p-cycles for every node and the results are shown in Table I and II. We can see that some nodes do not appear in Table I and II because there is not any node-encircling p-cycle when these nodes are seemed as the central nodes. We first evaluate the minimal-hops p-cycle C_{min} containing all the target nodes generated by WDCS and LFCA for every node of the topology in Table I. The nodes with the underline in the Table I denote that these nodes are the target nodes in the cycles. We can see that the LFCA can get cycle C_{min} containing less nodes than the WDCS when the nodes (4, 5, 10, 14, 20) are seemed as the central nodes. So the cycle C_{min} generated by LFCA is more suitable for the basic algorithms of the next step. When the target nodes are not the nodes which must be the neighbors of

TABLE I. COMPARISON OF WDCS AND LFCA FOR THE MINIMAL-COST P-CYCLE C_{MIN}

Algorithm	Central node 2	Central node 4	Central node 5	Central node 8	Central node 9	Central node 10	Central node 11	Central node 14	Central node 15	Central node 16	Central node 20
The cycle C_{min} of WDCS	**1-6-7-3-4-0-1**	**2-3-7-9-8-6-1-0-2**	**6-8-9-7-2-1-6**	**9-12-11-13-10-7-6-9**	**7-10-13-11-12-8-6-7**	**9-11-13-15-17-20-19-18-14-12-8-6-7-9**	**12-15-13-10-7-9-12**	**11-9-7-10-13-16-17-20-19-18-15-12-11**	**11-13-16-17-20-19-18-14-12-11**	**15-17-20-19-18-14-12-11-13-15**	**15-19-18-14-12-11-9-7-10-13-16-17-15**
The cycle C_{min} of LFCA	**1-6-7-3-4-0-1**	**2-3-7-6-1-0-2**	**6-7-2-1-6**	**9-12-11-13-10-7-6-9**	**7-10-13-11-12-8-6-7**	**9-11-13-15-12-8-6-7-9**	**12-15-13-10-7-9-12**	**15-18-19-20-17-16-13-11-12-15**	**11-13-16-17-20-19-18-14-12-11**	**15-17-20-19-18-14-12-11-13-15**	**15-19-18-14-12-11-13-16-17-15**

TABLE II. COMPARISON PERFORMANCE OF ENUMERATIO WDCS AND LFCA

Performance of algorithms	Central node 2	Central node 4	Central node 5	Central node 8	Central node 9	Central node 10	Central node 11	Central node 14	Central node 15	Central node 16	Central node 20
Number of candidate cycles for Enumeration	248	248	868	495	15	240	768	113	113	113	113
Number of candidate cycles for WDCS	36	18	30	11	11	22	48	20	26	28	19
Number of candidate p-cycles for LFCA	64	50	77	12	15	78	71	26	36	35	36
Average AE per candidate p-cycle of WDCS	1.741	2.183	2.015	1.874	1.469	2.261	1.841	2.234	1.646	2.047	2.231
Average AE per candidate p-cycle of LFCA	1.839	2.071	2.029	1.877	1.474	2.191	1.88	2.199	1.684	2.066	2.197

the central node, the cycles generated by LFCA are the simple link p-cycles. So the LFCA is suitable for finding both the node-encircling p-cycles and the link p-cycles. We compare the performance of Enumeration, WDCS and LFCA in Table II. When every node of the test network is supposed to be the central node and we use WDCS, LFCA and enumeration to compute the corresponding candidate node-encircling p-cycles. The numbers of candidate node-encircling p-cycles generated by three algorithms in the test networks are shown in the Table II. The average AE and hops per candidate cycle generated by LFCA and WDCS are shown in the latter of the Table II. These results show that the LFCA can find the simple node-encircling candidate p-cycles for any central node if such a simple node-encircling p-cycle exists in the network. The LFCA can get more candidate p-cycles with good efficiency than WDCS by performing the basic operation algorithms without enumerating all cycles.

V. CONCLUSIONS

In this paper, we proposed a new heuristic finding p-cycles algorithm called LFCA to find the node-encircling p-cycles in survivable mesh networks. LFCA can find more good candidate simple p-cycles in the mesh networks without enumerating all cycles. Compare to previous algorithms, LFCA can efficiently achieve node failure protection in survivable mesh networks.

REFERENCES

[1] W.D. Grover, D. Stamatelakis, Cycle-oriented distributed preconfiguration: ring-like speed with mesh-like capacity for self-planning network restoration, In Proc. of IEEE ICC 1998, June 1998, pp. 537-543.

[2] D. Stamatelakis, W. D. Grover, IP layer restoration and network planning based on virtual protection cycles, IEEE Journal on Selected Areas in Communications, vol. 18, no. 10, Oct. 2000, pp. 1938-1949.

[3] G. Shen, W. D. Grover, Extending the p-cycle concept to path segment protection for span and node failure recovery, IEEE Journal on Selected Areas in Communications, vol. 21, no. 8, Oct. 2003, pp. 1306-1319.

[4] K.W.Ko and K.W.Cheung, "A Local Map Based (LMB) Self-Healing Scheme for Arbitrary Topology Networks," Globecom'97, 1997,pp.1393-1397.

[5] Bin Wu. Yeung K.L. King-Shan Lui and Shizhong Xu. A New ILP-Based p-Cycle Construction Algorithm without Candidate Cycle Enumeration, IEEE International Conference on Communications (ICC 2007). June 2007, pp.2236 – 2241.

[6] Kang, B. Habibi, D. Lo, K. and etc. An approach to Generate An Efficient set of Candidate p-Cycles in WDM Mesh Networks, Asia-Pacific Conference on Communications (APCC 2006), Aug. 2006 pp.1-5.

[7] John Doucette, Peter Giese, Wayne Grover, Combined Node and Span Protection Strategies with Node-Encircling p-Cycles, Proceedings of the Fourth International Workshop on the Design of Reliable Communication Networks (DRCN 2005), Italy, October 2005, pp.213-221.

[8] Chang Liu and Lu Ruan. Finding Good Candidate Cycles for Efficient p-Cycle Network Design, ICCCN 2004. Chicago. IEEE. October 2004, pp. 321-326.

[9] Grover, W. D., and Doucette, J. E., "Advances in Optical Network Design with p-Cycles: Joint Optimization and Pre-Selection of Candidate pcycles," Proceedings of the IEEE-LEOS Summer Topical Meeting on All Optical Networking 2002, Mont Tremblant, Quebec, Canada, pp. 49-50.