A new heuristic routing algorithm with Hamiltonian Cycle Protection in survivable networks

Lei Guo a,b, Xingwei Wang a,*, Juan Du a, Tengfei Wu a

a College of Information Science and Engineering, Northeastern University, Shenyang 110004, China
b Key Laboratory of Broadband Optical Fiber Transmission and Communication Networks, University of Electronic Science and Technology of China, Chengdu 610054, China

Received 25 July 2007; received in revised form 2 November 2007; accepted 13 November 2007
Available online 22 November 2007

Abstract

In high speed backbone networks, the survivability is a very important issue since the networks carry a lot of traffic. In this paper, we study the protection method with Hamiltonian Cycle in meshed survivable backbone networks, and proposed a new heuristic routing algorithm called Enhanced Hamiltonian Cycle Protection (EHCP) to tolerate the single failure. In EHCP, for each connection request the working path can be effectively selected based on the presented link-cost functions that consider the different affection for backup resources from on-cycle links and straddling links so that the backup resources on the protecting Hamiltonian Cycle can be reduced. The simulation results show that, compared with the conventional Hamiltonian Cycle Protection algorithm, EHCP can obtain significant improvement in resource utilization ratio.

Keywords: Survivable networks; Routing; Protection; Hamiltonian Cycle; Heuristic algorithm

1. Introduction

With the rapid development of the next generation networks, the explosions of related services are creating a huge demand of bandwidth which also results in high quality of services required from customers. Therefore, the survivability has become a key issue in designing the backbone networks. The strategy of survivability is to ensure the networks with services continuity by preplanning the backup resources for working resources. In previous works [1–3], most authors studied the shared path protection in which each connection request can be equipped with a disjoint routed working and backup path pair. Once the working path is unavailable by unexpected failures, the working traffic can be switched to the corresponding backup path such that the service can be protected. Since the single-link failure is dominant in current backbone networks, e.g., high speed optical networks, this paper focuses on the protection design for this failure scenario [1–3].

Although the shared path protection method performs efficient resources utilization, it may lead to long restoration time, and also the protection switching procedure is complicated since failures on the path may lead to many messages and signals notifying the repaired source node. Therefore, the effective management for shared path protection is a challenge. To overcome the drawback, some works have proposed the
protection method based on $p$-cycles [4–11], which can achieve the efficient resources utilization as the shared mesh protection method meanwhile perform the fast restoration time as the ring networks. Generally, in order to provide effective protection with $p$-cycles, the network may be divided into many local areas that may lead to complicated cooperation of different cycles. As a special case of $p$-cycles, Hamiltonian Cycle Protection (HCP) method has been proposed which can achieve fast restoration and simple management [12–15]. In order to achieve HCP, there must be at least a Hamiltonian Cycle in the network. Fortunately, we can find the Hamiltonian Cycles in most current backbone networks, e.g., US National network, CHINA CERNET, NJLATA network, ECNET, etc. In the following, according to Fig. 1 we explain the basic idea of HCP.

In Fig. 1, there exists a Hamiltonian Cycle A–B–C–D–E–H–F–G–A, where the links in the network can be divided into two categories: on-cycle links and straddling links. Obviously, these links on Hamiltonian Cycle are on-cycle links (thick links in Fig. 1), and other links are straddling links (thin links in Fig. 1). In Fig. 1a, any single failure of on-cycle link can be protected by the residual available routes on Hamiltonian Cycle; for example, the failure of on-cycle link A–B can be protected by the route B–C–D–E–H–F–G–A. Therefore, the backup resources on Hamiltonian Cycle should be enough to protect the working resources on the failed link; for example, if there are four working units on link A-B, there will need four backup units on each link on Hamiltonian Cycle. Another case is in Fig. 1b, where any single failure of straddling link can be protected by two available routes on Hamiltonian Cycle; for example, the failure of straddling link H–D can be protected by the routes H–E–D and H–F–G–A–B–C–D. Therefore, the backup resources on Hamiltonian Cycle should be enough to protect the half of working resources on the failed link; for example, if there are six working units on link H–D, there will need three backup units on each link on Hamiltonian Cycle.

Although the HCP method is simple and effective to provide protection for any single failure of on-cycle link or straddling link, it may increase the redundant backup resources and result in bad resource utilization if the working paths of connection requests cannot be selected optimally, which can be seen in Section 2.2. Since this problem of HCP has not been solved by heuristic algorithms in previous works [12–15], in this paper we will propose a new fast heuristic routing algorithm called Enhanced Hamiltonian Cycle Protection (EHCP) to improve the resource utilization. We will present the link-cost functions that consider the different affection for backup resources from on-cycle links and straddling links such that the backup resources on the protecting Hamiltonian Cycle can be significantly reduced.

The rest of this paper is organized as follows: Section 2 states the system model, backup resources assignment, and link-cost functions; Section 3 proposes the processes of heuristic algorithm; Section 4 presents the simulations and analysis; Section 5 concludes this paper.

2. Problem statement

2.1. System model

We assume the given network has $N$ nodes and $L$ links, each connection request arrives at the network orderly, and there is only a connection request arrives at a time. The shortest path algorithm, i.e., Dijkstra’s algorithm, is applied to compute the routes. The following notations are introduced.

- $OL$: Set of on-cycle links which are traversed by Hamiltonian Cycle.
- $SL$: Set of straddling links which are not traversed by Hamiltonian Cycle.
- $WP_r$: Working path of connection request $r$.
- $RU_r$: Number of resource units required by connection request $r$.
- $TU_j$: Number of total resource units on link $j$.
- $WU_j$: Number of working units consumed on link $j$.
- $FU_j$: Number of free resource units on link $j$.
- $BU$: Number of backup units required on each link on Hamiltonian Cycle.
- $HL$: Number of links traversed by Hamiltonian Cycle.

2.2. Backup resources assignment

We assume there is only single-link failure in the network such that the backup units needed on each link on

![Fig. 1. Hamiltonian Cycle Protection: (a) Protecting on-cycle link; (b) protecting straddling link.](image-url)
Hamiltonian Cycle can be obtained according to (1). It is also obvious that the total number of units needed on Hamiltonian Cycle is equal to \( BU \cdot HL \).

\[
BU = \max \left\{ \max \left( WU_j | j \in OL \right), \max \left( \frac{WU_j}{2} | j \in SL \right) \right\}
\]

According to (1) we can see that the backup units are determined by the maximum value of working units on the on-cycle links and straddling links. Further, since there are two different protection paths on Hamiltonian Cycle that can be used to recovery the traffic on each straddling link, it will be efficient to only consider the maximum value of half of working units on the straddling links. Based on this point, when computing the working paths for each connection request, we can consider two issues for selecting the proper links: (1) select these links with more free resources based on the load balancing idea as far as possible; (2) select these links as far as possible if the selection will not lead to assign new backup resources. This idea presented here can be illustrated in Fig. 2 where there exists a Hamiltonian Cycle A–B–C–D–E–H–F–G–A.

In Fig. 2a, we assume that there has already existed some working units consumed on links A–B, B–C, A–H and H–D; that is, \( WU_{A,B} = 4, WU_{B,C} = 2, WU_{A,H} = 4, \) and \( WU_{H,D} = 4 \). The working units on other links are all equal to zero. It is obvious that for each link, more working units mean less free units. According to (1), we can easily obtain that \( BU = 4, HL = 8 \), and the total backup units consumed on Hamiltonian Cycle are \( BU \cdot HL = 32 \).

Assume a new connection request is from source node A to destination node C and is with four resource units required. The standard shortest path algorithm generally will select the route A–B–C as the working path in Fig. 2b, such that the working units consumed are changed to \( WU_{A,B} = 8, WU_{B,C} = 6, WU_{A,H} = 4, \) and \( WU_{H,D} = 4 \). According to (1), we can obtain that \( BU = 8, HL = 8, \) and the total backup units consumed on Hamiltonian Cycle are \( BU \cdot HL = 64 \). In Fig. 3b, therefore, the total consumed resource units including working units (=22) and backup units (=64) are equal to 86.

However, if considering the load balancing idea the total backup units consumed on Hamiltonian Cycle can be significantly saved. Based on load balancing, in Fig. 2c we encourage the working path to traverse these links that have less working units and more free resource units so that the load can be properly dispersed to different links for reducing the maximum value of working units on all links. Therefore, the working path is selected as A–G–F–H–E–D–C in Fig. 2c, such that \( WU_{A,H} = 4, WU_{H,D} = 4, WU_{B,C} = 2, \) and the working units consumed on other on-cycle links are all equal to 4. According to (1), we can obtain that \( BU = 4, HL = 8, \) and the total backup units on Hamiltonian Cycle are \( BU \cdot HL = 32 \). Therefore, in Fig. 2c the total resource units including working units (=38) and backup units (=32) are equal to 70 that is less 16 resource units than that of in Fig. 2b. It is obvious that the load balancing method can save resources units compared with the
shortest path method. However, this can be further improved by selecting some proper links.

In Fig. 2d, we encourage the working path to traverse these links as far as possible if the backup resource units can be saved. Thus, the working path can be A–H–D–C that contains two straddling links A–H and H–D as shown in Fig. 2d, such that the working units consumed are $WUA_B = 4$, $WUB_C = 2$, $WUA_H = 8$, $WUD_C = 4$. According to (1), we can obtain that $BU = 4$, $HL = 8$, and the total backup units on Hamiltonian Cycle are $BU \cdot HL = 32$. Therefore, in Fig. 2d the total consumed resource units including working units (=26) and backup units (=32) are equal to 58 that is less 12 resource units and 28 resource units than those of in Fig. 2b and c, respectively.

It is obvious that the presented method that considers both load balancing and encouragement of selecting proper links can save much more resources units. Therefore, we need to design link-cost functions in computing the working paths for the routing algorithm (i.e., Dijkstra’s algorithm) to achieve the reduction of total resource units by load balancing and encouragement of selecting the proper links as far as possible.

2.3. Link-cost functions

Based on our presented policy in Section 2.2, for a connection request $r$ the cost for any link $j$ can be defined as:

$$\text{Cost}_j = \begin{cases} +\infty, & \text{if } (FU_j < RU_j) \\ \frac{FU_j + 1 - RU_j}{FU_j} \cdot C_j, & \text{if } (FU_j \geq RU_j) \end{cases}$$

In (2), the $C_j^r$ can be further written as (3), where $\sigma$ is constant factor that is smaller than 1.

$$C_j^r = \begin{cases} +\infty, & \text{if } (j \in \text{OL}, WU_j + RU_j > BU + FU_j, \exists f \in \text{HC}) \\ \text{or } (j \in \text{SL}, \frac{WU_j + RU_j}{2} > BU + FU_j, \exists f \in \text{HC}) \\ 1, & \text{else if } (j \in \text{OL}, WU_j + RU_j > BU) \text{ or } (j \in \text{SL}, \frac{WU_j + RU_j}{2} > BU) \\ \sigma, & \text{else if } (j \in \text{OL}, WU_j + RU_j \leq BU) \text{ or } (j \in \text{SL}, \frac{WU_j + RU_j}{2} \leq BU) \end{cases}$$

We can see that in (2) the costs of links that have no enough resource units will be set to infinite while the costs of links that have enough resource units will be set to finite values according to load balancing idea; that is, more free resource units mean less link-costs. In (3), we can see that there are three cases for link-costs assignments:

(1) Case 1: if the working path traverses link $j$ and the sum of free and backup units of some link $f$ on Hamiltonian Cycle will not be enough, link $j$ will have infinite cost.
(2) Case 2: if the working path traverses link $j$ and the new backup units of any link on Hamiltonian Cycle will need to be assigned, link $j$ will have cost of value 1.
(3) Case 3: if the working path traverses link $j$ and the new backup units of any link on Hamiltonian Cycle will not need to be assigned, link $j$ will have cost of value $\sigma$ (0 < $\sigma$ < 1).

Therefore, it is obvious that the link-costs assignments in (2) and (3) can effectively encourage the load balancing and proper links selection to reduce the consumed resource units.

![Diagram](image_url)

Fig. 3. Test networks: US National, CHINA CERNET, NJLATA, and ECNET.
Fig. 4. Performances of resource utilization ratio in: (a) National network; (b) CERNET network; (c) NJLATA network; (d) ECNET network.
3. Heuristic algorithm

The processes of the proposed heuristic algorithm EHCP can be presented as follows:

**Input:** Network topology information; \( Q \) connection requests; \( r \leftarrow 0 \).

**Output:** Total resource units consumed.

**Step 1:** Find the Hamiltonian Cycle by some off-line manner (e.g., deep-first-search method) for the given network, calculate \( HL \), and let \( BU \leftarrow 0 \).

**Step 2:** If \( r \geq Q \), go to Step 6; otherwise, go to Step 3.

**Step 3:** Adjust the link-costs according to (2) and compute the least-cost working path \( WP_{r} \). If \( WP_{r} \) has been found, go to Step 4; otherwise, go to Step 5.

**Step 4:** Record \( WP_{r} \), let \( WU_{j} \leftarrow WU_{j} + RU_{j}(\forall j \in WP_{r}) \), calculate and update \( BU \) according to (1), set \( r \leftarrow r + 1 \), and go backup to Step 2.

**Step 5:** Block this connection request, set \( r \leftarrow r + 1 \), and go back to Step 2.

**Step 6:** Return the total resource units consumed, i.e., \((BU \cdot HL + \sum_{j \in HL} WU_{j})\).

In the processes above, the Hamiltonian Cycle can be obtained by some off-line manner for the given network topology in Step 1, so the time complexity for this can be ignored. Here, we mostly consider the time complexity of computing the working path for each connection request. In this worst case, for each connection request the time complexities from Step 2 to Step 6 are \( O(1) \), \( O(L + N^{2}) \), \( O(2 + 2L) \), \( O(1) \), and \( O(L) \), respectively. Therefore, the time complexity of EHCP in this worst case is approximately \( O(4 + 4L + N^{2}) \) that is obvious a very fast heuristic method.

4. Simulations and analysis

We simulate an incremental traffic model. In this model, the matrix of connection requests is not known ahead of time. Each new connection request enters the network one by one. Once allocated, connection requests in the network cannot be reconfigured. The granularity of required bandwidth of each connection request can be selected as 1, 2, and 3 U. Also, we assume that there are no waiting queues in the network, so if a connection request was blocked, it would be abandoned immediately. The test networks are US National, CHINA CERNET, NJLATA, and ECNET as shown in Fig. 3, where there exist Hamiltonian Cycles which are shown as thick lines for all of the networks. In these networks, each link is assumed to have 200 resources on it. The computer used in simulation is configured with 2.8 GHz CPU and 1G DDRRAM and the software is VC++6.0. We compare the performances of resource utilization ratio (RUR) of EHCP with the conventional HCP that only considers the load balancing routing. The RUR is the ratio of the total consumed backup units over the total consumed working units. Smaller value of RUR means the better performance of resource utilization ratio.

In Fig. 4, the X in HCP-X or EHCP-X denotes the granularity of required bandwidth. We can clear see that, with different granularities, the resource utilization ratios of EHCP are better than those of HCP in all National, CERNET, NJLATA, and ECNET networks, and the improvement of EHCP over HCP is up to 50% which is significant and promising. The reasons for this is that EHCP considers the different affection for backup resources from on-cycle links and straddling links, and it encourages the working path as far as possible to traverse these links that will lead to less backup resources assignment on the protecting Hamiltonian Cycle. Therefore, the total resources consumed can be reduced and the resource utilization ratio of EHCP can be significant better than that of HCP.

5. Conclusions

This paper has investigated the problem of Hamiltonian Cycle protection for the single failure in survivable networks, and proposed a new heuristic routing algorithm called EHCP. In EHCP, we have designed the link-cost functions that consider the different affection for backup resources from on-cycle links and straddling links to reduce the backup resources on the protecting Hamiltonian Cycle. Compared with the conventional HCP algorithm, EHCP can obtain significant improvement in resource utilization ratio.

Acknowledgement

The authors like to thank reviewers for valuable comments.

References


Lei Guo was born in Sichuan, China, in 1980. He received the Ph.D. degree in communication and information systems from School of Communication and Information Engineering, University of Electronic Science and Technology of China, Chengdu, China, in 2006. He is currently an Associate Professor in College of Information Science and Engineering, Northeastern University, Shenyang, China. His research interests include survivability, optical networking, and next generation internet. Dr. Guo is a member of IEEE and a member of OSA. He was the recipient of the Best Paper Award from the International Conference on Communications, Circuits and Systems (ICCCAS’04). Dr. Guo is currently servicing as an Editorial Board Member of The Open Optics Journal.

Xingwei Wang received the B.Sc., M.Sc., and Ph.D degrees in computer science from Northeastern University, China, in 1989, 1992, and 1998, respectively. He is currently a Professor in the College of Information Science and Engineering, Northeastern University, China. His research interests are mainly on routing algorithms and protocols, QoS (Quality of Service) control schemes, fault-tolerance and survivability models, mobility management mechanisms and resource assignment methods in NGI (Next Generation Internet), IP over DWDM optical Internet and mobile Internet. He has published over 100 technical papers in the above areas.

As one of the co-investigators, he has won the first-level science and technology advancement award of China ministry of education twice.

Juan Du was born in Liaoning, China, in 1985. He received the B.S. degree in communication engineering from College of Information Science and Engineering, Northeastern University, Shenyang, China, in 2006. She is currently a postgraduate student at the same university. Her research interests include optical networking, multi-layer networks and survivability.

Tengfei Wu was born in Liaoning, China, in 1984. He received the B.S. degree in communication engineering from College of Information Science and Engineering, Northeastern University, Shenyang, China, in 2005. He is currently a postgraduate student at the same university. His research interests include optical networking, waveband switching, routing and survivability.