A Unified Approach to Routing Protection in IP Networks

Qi Li, *Member, IEEE,* Mingwei Xu, *Member, IEEE,* Jianping Wu, *Fellow, IEEE,* Patrick P. C. Lee, *Member, IEEE,* Xingang Shi, *Member, IEEE,* Dah Ming Chiu, *Fellow, IEEE,* and Yuan Yang, *Student Member, IEEE*

***Abstract*—Routing failures are common on the Internet and routing protocols can not always react fast enough to recover from them, which usually cause packet delivery failures. To address the problem, fast reroute solutions have been proposed to guarantee reroute path availability and to avoid high packet loss after network failures. However, existing solutions are often specific to single type of routing protocol. It is hard to deploy these solutions together to protect Internet routing including both intra- and inter-domain routing protocols because of their individual computational and storage complexity. Moreover, most of them can not provide effective protection for traffic over failed links, especially for the bi-directional traffic. In this paper, we propose a unified fast reroute solution for routing protection under network failures. Our solution leverages identifier based direct forwarding to guarantee the effectiveness of routing protection and supports incremental deployment. In particular, enhanced protection cycle (*e-cycle*) is proposed to construct rerouting paths and to provide node and link protection for both intra- and inter-domain routing protocols. We evaluate our solution by simulations, and the results show that the solution provides 100% failure coverage for all end-to-end routing paths with approximately two extra Forwarding Information Base (FIB) entries. Furthermore, we report an experimental evaluation of the proposed solution in operational networks. Our results show that the proposed solution effective provides failure recovery and does not introduce processing overhead to packet forwarding.**

***Index Terms*—IP networks; routing; routing protection; resilience.**

Manuscript received April 9, 2011; revised October 5, 2011, January 25 and March 28, 2012; accepted April 6, 2012. The associate editor coordinating the review of this manuscript and approving it for publication was J. Schönwälder.

This work was supported by the National Natural Science Foundation of China under Grant No. 61073166, Grant No. 61133015, and Grant No. 61161140454; by the National Basic Research Program of China (973 Program) under Grant No. 2009CB320502 and Grant No. 2012CB315803; and by the National High-Tech Research and Development Program of China (863 Program) under Grant No. 2011AA01A101. The preliminary version of this paper titled "Achieving Unified Protection for IP Routing," was published in the *Proceedings of the 19th International Conference on Computer Communications and Networks* (ICCCN), 2010 [28]. This journal version makes the following extensions to the conference paper: (i) we specifically address BGP (iBGP/eBGP) node protection, which is not addressed in the literature; (ii) we evaluate the performance of different existing protection solutions in the GEANT and Rocketfuel networks, and compare the overhead of these solutions; (iii) we discussed the implications for real deployment of *e-cycle*, and demonstrate the practicality of *e-cycle* by deploying it in operational networks.

Q. Li, M. Xu, J. Wu, and Y. Yang are with the Department of Computer Science, Tsinghua University, Beijing, China, 100084 (e-mail: {liqi, xmw, jianping, yyang}@csnet1.cs.tsinghua.edu.cn).

P. Lee is with the Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong (e-mail: pclee@cse.cuhk.edu.hk).

X. Shi is with the Network Research Center, Tsinghua University, Beijing, China, 100084 (e-mail: shixg@cernet.edu.cn).

D. Chiu is with the Department of Information Engineering, The Chinese University of Hong Kong, Hong Kong (e-mail: dmchiu@ie.cuhk.edu.hk).

Digital Object Identifier 10.1109/TNSM.2012.070512.110138

I. Introduction

INTERNET routing connects different IP networks and plays a critical role in ensuring packet delivery throughout the Internet. However, previous studies show that current routing systems are ineffective in recovering from routing failures. For instance, the 2006 earthquake in Taiwan caused global disruption in the Internet although there remained unaffected links that could provide potential connectivity for different IP networks. Even though most routing failures, such as Border Gateway Protocol (BGP) session resets and transient hardware failures, are short-term and last less than 3 minutes [1], current routing protocols fail to react quickly to recover from such short-term routing failures. It is not unusual for routing protocols to take several minutes or even longer to converge [2]. This significant recovery time leads to unreliable packet delivery.

Extensive research has been conducted in order to effectively address routing failures. One line of work is to develop solutions that provide *fast routing convergence*, which has been extensively studied in the literature [3], [4]. However, none of these solutions has been deployed in operational networks due to their complexity, or in some cases, due to subtle design flaws. For instance, Ghost Flushing [3] expedites convergence by sending extra route withdrawal messages but may exacerbate routing convergence in failover events. Basically, fast routing convergence is not effective for handling routing blackholes and loops.

Another line of work that addresses routing failures is to realize routing protection by using backup routing paths [5], [6], [7], [8], [9], [10], i.e., *fast reroute approaches*. However, such approaches again have different design limitations. IP-FRR solutions [6], [7], which are active subjects in the IETF, focus only on the protection of intra-domain routing. They share important drawbacks such as difficult deployment and/or uncertain protection effectiveness over failed links [11]. To support fast reroute in inter-domain routing, Bonaventure *et al.* [5] propose BGP fast reroute (BGP-FRR), the first solution that protects external BGP (eBGP) between different ASes by automatic protection and aims to realize effective protection by extra manual configurations. R-BGP [12] is proposed to provide automatic failover for eBGP failures. R-BGP requires an extra Forwarding Information Base (FIB) entry for every prefix under protection, and the protection effectiveness is greatly restricted by routing policies. Also, neither BGP-FRR nor R-BGP considers internal BGP (iBGP) failures [13], [14].

Furthermore, previous studies consider protection only for a single type of routing protocol, either intra- or inter-domain routing. It would be a difficult task to deploy these solutions

together to protect Internet routing in operational networks which consists of both intra- and inter-domain routing protocols. Thus, a light-weight *unified* routing protection solution is essential to practically improve protection effectiveness and achieve real deployment.

In this paper, we propose a unified routing protection solution to detour failures and realize fast rerouting for different types of routing, especially for iBGP and eBGP. Our key observation is that both intra- and inter-domain routing protocols are highly correlated and a protection solution should not separate them. Also, by employing a unified solution, we may greatly reduce the number of Forwarding Information Base (FIB) entries. In our solution, we propose enhanced protection cycle called *e-cycle*, a fast reroute solution that constructs effective rerouting paths for node and link protection in both intra- and inter-domain routing protocols. The protection effectiveness of the solution is not restricted by routing policies. Moreover, our solution is independent of specific routing protocols and is incrementally deployable.

The contributions of this paper can be summarized as follows:

- We propose *e-cycle*, a practical and unified solution that effectively addresses routing failures in both intra- and inter-domain routing protocols, and *e-cycle* is lightweight and incrementally deployable.
- Using simulation, we evaluate both *e-cycle* and existing solutions for routing failures under real-life network topologies, and we show that *e-cycle* provides 100% failure coverage in both intra- and inter-domain routing protocols.
- We implement *e-cycle*, and report a preliminary experimental study that shows the empirical performance of *e-cycle* under partial deployment in operational networks. Our goal is to show the practicality of *e-cycle* in actual deployment. We also explore the important implications of routing protection from our experiments, which are not well addressed in previous studies.

The rest of the paper is organized as follows. Section II identifies the drawbacks of existing fast reroute solutions. We introduce our *e-cycle* solution and propose different algorithms to construct *e-cycle* in Section III, and evaluate the performance of our solution in Section IV. We report an experimental study of our solution in Section V. Section VI concludes this paper and suggests future work.

II. Problems in Existing Fast Reroute Solutions

A. Traditional Intra-Domain Fast Rerouting

Several fast reroute solutions are proposed to forward packets along an alternate path under network failures, and then to provide routing protection and improve routing performance [6], [15]. Failure insensitive routing (FIR) [16] is proposed to provide routing protection by interface specific forwarding. Congestion and performance predictability during rerouting are also addressed [17], [18]. However, most solutions can not provide assured protection effectiveness and is not deployed in practice due to the computational complexity.

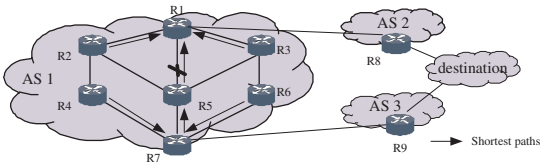


Fig. 1. Fast reroute to network failure.

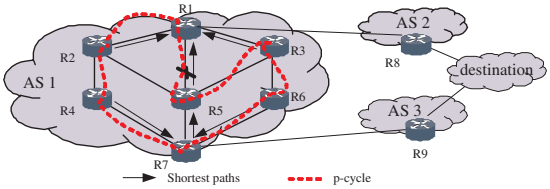


Fig. 2. Fast Reroute with *p-cycle*.

Among them, the *Not-via* approach [15] provides the best performance of failure coverage among various IP fast reroute (IP-FRR) solutions in intra-domain routing [15], [11], [19], [16], [20], [21], [22], [23], [24].

In Not-via, to recover from a node or link failure, the router adjacent to the failure will attempt to deliver packets with a precomputed protection path. The router will encapsulate the packets with a special Not-via address indicating that packet forwarding is **not via** the failed component. The packets will firstly be forwarded to the decapsulation point, i.e., the router on the opposite side of failed component, and then the packets will be decapsulated and forwarded by normal routes.

Node protection is recommended to detour failures and reduce the computational complexity [15], but special consideration is required for some corner cases. For instance, as shown in Figure 1, packets at R7 are forwarded towards R1. If link R1-R5 fails and node protection for R1 is activated to protect the link, then R5 will encapsulate the packets with a new IP header using a special not-via address as the destination address, such that these packets will **not** be routed **via** R1. Unfortunately, if the original destination of these packets is R1, then it is impossible to find a rerouting path to R1 not via R1 by node protection.

The problem above can be solved by applying *link protection* instead of node protection. That is, we can use the not-via address to indicate that packets to R5 are not via R1 to forward the packets **not via** link R1-R5. This would provide more effective protection at the cost of more computations for not-via addresses, e.g., in R5. Thus, it is obvious that it will introduce more overhead to compute and store extra FIB entries for all Not-via addresses. To address this issue, the concepts of multiple topologies and redundant trees are applied to reduce the computation cost in [25], [26], [27]. Unfortunately, the storage overhead in these approaches is still not effectively reduced.

B. Inter-Domain Fast Rerouting

In addition to the above issues, intra-domain routing failures may trigger re-computation of inter-domain (i.e., BGP) routes. For example, as shown in Figure 5, R1 and R8 build an eBGP

session, and R7 and R9 build an eBGP session. Then, R1 and R7 build an iBGP session to advertise their learned eBGP routes. If the protection for link R1-R5 fails, then iBGP control messages between R1 and R7 will be dropped and the BGP session will be eventually broken. Thus, R1 in AS 1 will select AS 2 instead of AS 3 as the next hop to the destination, and all descendant ASes of R1 in AS 1 will have to recompute their routes to the destination. Note that configuring BGP sessions with loopback interfaces still cannot solve such problem. If protection for link R1-R5 fails and all packets between R1 and R7 will be dropped, BGP session between R1 and R7 will eventually expire.

Although several approaches have been proposed to address BGP protection, they focus on eBGP protection only and are unable to protect such iBGP failures. Bonaventure *et al.* proposed an automatic solution called BGP-FRR specifically for external BGP (eBGP) protection, and provide different protection strategies for different multi-homing stub networks [5]. Francois *et al.* [28] preactivate alternate routing paths for recovery from manual shutdown of eBGP peering links and prevent packet loss from failure caused by maintenance operations. Kushman *et al.* proposed an improved BGP, R-BGP [12], in which several failover paths are pre-computed and stored in BGP RIBs, and failover paths will take effect if failures are detected. R-BGP requires adding an FIB entry for every protected prefix in each router. Bryant *et al.* presented an approach to implement inter-domain routing with Not-via addresses [15]. However, they do not consider the routing policy issue, e.g., it may not be possible for an AS to provide failure detour for their provider ASes since it requires the AS to pay for network connectivity for their providers. Moreover, these BGP protection solutions cannot provide failure detours for iBGP failures even though they introduce considerable computational and management overhead. If we consider protecting both intra- and inter-domain routing using these solutions, the number of extra FIB entries may be much more than existing FIB entries.

C. Virtual Cycle Based Routing Protection

Virtual protection cycle (*p-cycle*) [29], [30] provides a practical and lightweight solution for routing protection, and it is first designed for failure recovery in SONET and WDM. In the *p-cycle* approach, minimal candidate virtual cycles are constructed to provide fast reroute for node and link failure recovery [30]. Thus, *p-cycle* only requires very few forwarding entries for efficient routing protection. Figure 2 depicts the principle of *p-cycle* for node and link protection. A *p-cycle* is pre-configured as a closed cycle (R1-R5-R3-R6-R7-R4-R2) which protects both on-cycle and straddling (off-cycle) failures [30]. Upon a failure of link R1-R5, *p-cycle* offers protection by the route on part of the remainder cycle (R5-R3-R6-R7-R4-R2-R1). The advantage of *p-cycle* is that it provides protection for all nodes and links with a few extra FIB entries and flexibly handles multiple failures [31]. The *p-cycle* approach would require no more than $2d$ additional FIB entries at each router for one type of routing protocol, where d is the number of neighboring routers to which it has direct links. This is a very small overhead compared to the typical number of FIB entries in other routing protection solutions [15].

However, *p-cycle* has some drawbacks. The length of a rerouting path in the original *p-cycle* solution is *significantly enlarged* under failures, as packets have to go through the whole remainder of the cycle and then be forwarded based on normal routes. For example, in Figure 2, *p-cycle* detours packets along the whole remainder of the cycle (R5-R3-R6-R7-R4-R2-R1) to offer protection from the failure of R1-R5. However, unlike nodes in WDM and SONET, in the IP network setting, routers within an AS have the entire intra-domain topology and packets should not be forwarded along such a long detour, e.g., R3 should be able to forward packets directly to R1 on behalf of R5.

Although Stamatelakis *et al.* adopted *p-cycle* in IP networks [31], their adoption requires modifications in current forwarding logic in each router [32]. The approach computes the cost of *every* packet to destinations during packet forwarding and the mechanism cannot be enabled in current router architecture. To address this issue, Cicic *et al.* proposed a per-node mapping approach to realizing early exit of p-cycle packets [32]. However, per-node mapping introduces much overhead to store the mapping records for all destinations. Furthermore, these approach adopting *p-cycle* may not provide protection for inter-domain routing. In inter-domain routing, Autonomous Systems (AS) in the Internet are operated by different ISPs with different routing policies. It may not be easy to deploy a p-cycle in the Internet since it requires configuring cycles on all nodes on the cycle. Thus, it is much more complex and harder to deploy a single *p-cycle* with the same *p-cycle* identifier on routers in different ASes which are usually operated by different ISPs.

III. E-CYCLE: ENHANCED PROTECTION CYCLE FOR ROUTING PROTECTION

In this section, we present the design of *e-cycle*, a solution using enhanced protection cycles for routing protection. We first present the overview of *e-cycle* and then propose different detailed algorithms to build *e-cycle* for different routing protocols.

A. Overview of e-cycle

Different from traditional auto-discovery protection solutions which introduce high deployment complexity to routing protocols [5], [15], [12], *e-cycle* provides efficient pre-configured routing paths to realize fast rerouting. Similar to *p-cycle* [31], *e-cycle* leverages virtual cycles to construct rerouting paths and uses different identifiers called *e-cycle IDs* (see discussion below) to uniquely identify these virtual cycles, thus provides protection for all nodes and links. The main difference is that, since every router has routes to destinations, *e-cycle* does not detour packets along an entire virtual cycle as in *p-cycle*, but seeks to find an earlier decapsulation point after which packets are again forwarded along normal routes. To achieve this, *e-cycle* introduces two components, namely, *protection initiators (PIs)* and *protection terminators (PTs)*. Protection initiators (PIs) are routers that detect failures and then activate protection paths to forward packets, and protection terminators (PTs) are routers that terminate protection

paths and continue normal packet forwarding. If a router detects a failure, then it will activate itself to become a PI, and will select a corresponding PT. We will discuss the PT selection in the following discussion.

The main idea of our *e-cycle* approach is that when an *e-cycle* is constructed, we select a PT for every PI in the cycle and packets are only forwarded along the partial cycle between the PI and PT. When a PI detects a failure, it starts to forward affected packets along the *e-cycle* towards its corresponding PT. Since we want to introduce as little overhead as possible to realize routing protection, we propose label *e-cycle ID* based direct forwarding. An *e-cycle* ID specifies the unique identifier of the *e-cycle* (i.e., virtual cycle) that is used for rerouting. It can be manually configured in routers that have deployed *e-cycle*, or distributed through an automatic mechanism such as Label Distribution Protocol (LDP) as in Not-via [15] [1]. To specify the correct *e-cycle* ID (and hence the *e-cycle*) in packets to be forwarded, each PI can simply encapsulate the packets with a new IP header using IP encapsulation (e.g., L2TP [33]) to keep backward compatibility, and the new IP header will contain an *e-cycle* ID field. In addition, we also include a *hop count* field in the new IP header. The hop count field specifies the hop count between PI and PT. It is used to indicate the lifetime of a packet in the *e-cycle*, and will be decremented by one when the packet is forwarded by a router. If the hop count equals to zero, then the packet will be removed from the *e-cycle* by the PT and the original packet will be forwarded along a normal route to its destination [2].

Figure 3 illustrates how *e-cycle* addresses the same failure as in Figure 2. Assuming R5 as a PI, we can choose R3 as the PT for R5 because the route to R1 in R3 will not pass though R5. R3 removes the *e-cycle* header and forwards it normally, and the length of the rerouting path in *e-cycle* is only 2. Thus, we can achieve an effective lightweight protection for intra-domain routing and further provide connectivity between iBGP speakers. To provide protection for eBGP, it is important to note that the *e-cycle* approach does not require all nodes in the cycle to have deployed *e-cycle* and to be configured with the same *e-cycle ID*. As shown in Figure 3, we assume that AS2 and AS3 are two provider ASes of AS1 and a virtual cycle (R1-R3-R6-R7-R9-⌊⌋-R8) (⌊⌋ denotes a sequence of traversed routers in which we do not need to configure *e-cycle* for eBGP protection) has been constructed. When link R1-R8 fails, R1 will detour packets along R3-R6-R7 to R9 and R9 definitely has routes to destinations.

Note that, in Figure 3, both AS 2 and AS 3 are provider ASes of AS 1, and then they can provide transit service for AS 1. For each destination, AS 2 and AS 3 can help AS 1 deliver packets to it. For any one link failure, e.g., link AS1-AS3 fails,

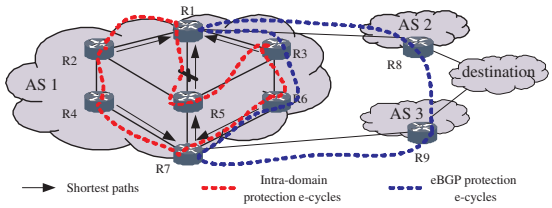


Fig. 3. Virtual cycles to recover from network failure.

AS 2 can help AS 1 deliver packets under failure with eBGP link AS2-AS3. If AS 1 has a routing reconvergence process after the failure, it will eventually select the route with the eBGP link AS2-AS3 as the best one. In this sense, it will not violate the routing policies. For the reverse traffic from AS 3 to AS 1, the situation is similar. For the failure of link AS1-AS3, if AS 2 and AS 3 deploy an e-cycle, AS 3 can deliver the packets to AS 1 with the help AS 2. However, if AS 2 and AS 3 do not have negotiation between themselves, AS 3 will still deliver packets to AS 1 with IP header. Normally, the IP addresses of eBGP link in R3 and R8 are managed by the same ISP (/AS), AS 1 or AS 2, and the addresses are known by both two ASes. Thus, we can safely encapsulate the packets to AS 1 with R8's IP address such that the encapsulated packets will be delivered to AS 2 according to the routing tables. After the packets reach R8, R8 will decapsulate the packets and send the packets to R3 in AS 1 eventually. Note that, the agreement between AS 2 and AS 1 may be required to protect the traffic from AS 1 to AS 3 over link AS1-AS3. However, AS 3 does not need to build agreements with AS 2 to protect traffic from AS 3 to AS 1 because the destination of the traffic to AS 1 will be encapsulated with R8's address which connects AS 2. AS 2 can directly forward the packets to AS 1, while not requiring decapsulating them for AS 1. The network configurations conform to the normal network operation practice. Similarly, if we enable routing re-convergence after link failure, the packets to AS 1 will finally get to AS 2 and AS 2 delivers the packets to final destinations. The difference between *e-cycle* and traditional routing reconvergence scheme is that *e-cycle* provides fast rerouting to deliver packets without routing reconverngence involved.

There are several types of failures that *e-cycle* must handle. For link failures, the failed link may or may not lie on the pre-configured *e-cycle*, and for node failures, the adjacent router may or may not lie on the same *e-cycle* as the failed one. Thus, our *e-cycle* solution should still be able to handle all these conditions by detouring packets to the corresponding PT as long as an *e-cycle* is pre-configured on a PI. Thus, we expect that *e-cycle* provides much better efficiency by realizing a unified protection for both node and link failures. Given that intra- and inter-domain routing protocols have different forwarding features, we should have different *e-cycle* construction methods for the protocols. In the following subsections, we will discuss two main issues: (i) how to construct *e-cycles*, and (ii) how to select the PT, in order to protect routing failures for different types of routing protocols.

[1]Normally, we only need to configure an e-cycle ID in each router. Compared to the tasks of configuring serial ports with IP addresses in each router, the management overheads are negligible. Thus, we suggest manually configure labels in routers. To realize label distribution, we need to extend link state advertisement (LSA) to piggyback label information and then different routers will learn which label they should use.

[2]Actually, hop count field we presented here is only to illustrate our design. In real implementation, we can re-use TTL to realize hop count field, and TTL should be set to the hop count value plus one. If we re-use TTL field, the e-cycle packets will be removed from the e-cycle by the PT when TTL equals to one and then the original packet will be forwarded along a normal route to its destination.

B. Intra-Domain Routing/iBGP Link Protection

We first describe how to provide link protection to intra-domain routing using *e-cycle*. Since iBGP relies on intra-domain routing, if we can guarantee link protection in intra-domain routing, then iBGP link failures can be eliminated. Note that the protection for an iBGP node is more complex because the node may be the only one egress point within an AS, and intra-domain protection can not successfully provide failure recovery. We will discuss this in Section III-D. So next we only discuss *e-cycle* construction for intra-domain routing and iBGP link protection. We assume that intra-domain routing uses shortest path routing (e.g., OSPF and IS-IS), and each router has formed a shortest path tree (SPT) that specifies all routing paths to other nodes within the domain. Also, given that virtual cycle construction in IP networks is well studied in the literature [31], [30], we leverage the same construction algorithm as *p-cycle* to construct virtual cycles. In the following discussion, we focus on the PT selection in the virtual cycles that have been constructed.

Algorithm 1 shows the intra-domain *e-cycle* construction algorithm. It returns a set C, in which each member c is a virtual cycle composed of the set of routers V_c in the cycle and the corresponding set of PTs S_c. First, we construct candidate virtual cycles using existing algorithms [31], [30](step 1). Then for each cycle c, we choose a PT for each router R_i^c on c (step 4-19). Note that c is uni-directional, and the nodes in V_c are ordered (in a cycle) as [R_1^c, ..., R_{i-1}^c, R_i^c, R_{i+1}^c, ..., R_m^c] such that when we traverse the cycle starting from R_i^c, R_{i+1}^c is the next node to be encountered and R_{i-1}^c is the last one. In the shortest path tree (SPT) rooted at R_i^c, if $SPT\_Desc(\mathrm{R}_i^c)$ returns the descendants of the subtree under the failed link (or failed node), then we try to find a router R_x^c in the cycle c, such that the shortest path from R_x^c to any router R_y in $SPT\_Desc(\mathrm{R}_i^c)$ does not pass R_i^c. That is, if R_i^c uses R_x^c to detour the failed link and forward packets to its descendent routers, then R_x^c will never send the packets back to R_i^c since the cost from R_x^c to R_y should be less than that from R_i^c to R_y. If such a router is found, then we can directly set R_x^c as the PT of R_i^c in the cycle c (step 5-16). Otherwise, the router R_{i+1}^c next to R_i^c along the cycle c will be chosen as the PT (step 17-19). For an *e-cycle*, a PI only needs one PT. In Algorithm 1, we will choose PT for PI if PT can be used to detour the failure and forward traffic to all destinations. In the worst case scenario, PT is the opposite to PI in the assumed failed link.

We now explain Algorithm 1 with an example. Figure 4 shows an intra-domain topology where the link weights are all set to 10, except that the weight of R5-R3 is 11. According to the link weights, all shortest paths root at different nodes are determined. For example, the shortest path tree rooted at R5 is shown in Figure 5(a). In the example, we assume that links R1-R5 and R2-R6 in the network fail.

We assume that two virtual cycles indicated by the dotted cycles are already constructed for these routers (as in step 1 of Algorithm 1), and we now illustrate how to choose a PT for each router in each cycle. For example, in Figure 4, we choose R3 as the PT for R5 in the directed cycle (R5-R4-R3-R2-R1) because the shortest paths from R3 to R5's SPT

Algorithm 1 Intra-domain *E-cycle* Construction

//SPT_Desc(R): the descendants of the subtree under the failed link (or failed node) in the SPT rooted at R;
//$SPT\_traversed$(R_x, R_y): the set of routers along the shortest path from R_x to R_y.
Input: intra-domain topology;
Output: $C = \{c | c = (V_c, S_c)\}$;

```
1: construct virtual cycles C={c|c = (V_c, ∅)};
2: for each c ∈ C do
3:    for each R_i^c ∈ V_c do
4:       flag = true;
5:       for (R_x^c in [R_{i+1}^c, R_{i+2}^c, ..., R_m^c, R_1^c..., R_{i-1}^c]) do
6:          for each R_y in SPT_Desc(R_i^c) do
7:             if (R_i^c ∈ SPT_traversed(R_x^c, R_y)) then
8:                flag = false;
9:                break;
10:            end if
11:         end for
12:         if (flag == true) then
13:            update_PT(c, R_i^c, R_x^c);
14:            break;
15:         end if
16:      end for
17:      if (flag == false) then
18:         update_PT(c, R_i^c, R_{i+1}^c);
19:      end if
20:   end for
21: end for
```

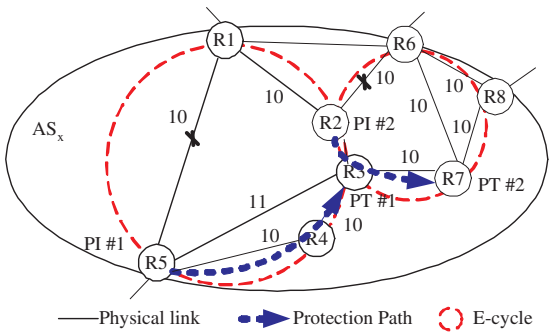
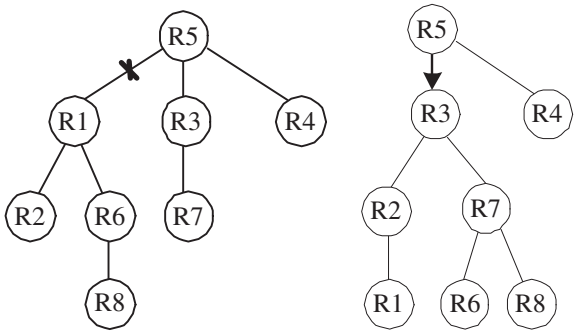


Fig. 4. *E-cycles* in intra-domain routing protection.

descendant nodes under the failed link R5-R1, such as R1, R2, R6 and R8, will not pass R5. However, R4 can not be used as R5's PT since the shortest path from R4 to R1 is R4-R5-R1 that includes the failed link R5-R1. For illustration, the shortest path tree rooted at R5 before and after the failure of link R1-R5 are depicted in Figure 5(a) and Figure 5(b), respectively. Once a PT is chosen, we need to distribute the alternate forwarding entries to the routers on this cycle for identifying the *e-cycle* ID. Figure 4 shows that we construct two *e-cycles* for eight routers in the AS. If any router detects a failure, then it can launch the protection with a specific PT in the cycle. For example, as shown in Figure 4, R5 acting as the PI activates the protection path to R3 once it detects the failure on R1-R5, and R2 activates the protection path to R7 once it detects the failure on R2-R6. In this way, traffic for R6 will go through R5, R4, R3, R2, R3 and R7, and finally be forwarded to R6 using normal route by R7.



(a) before failure (b) after failure

Fig. 5. The change of shortest path tree before/after network failure.

C. eBGP Link Protection

External BGP (eBGP) protection is different from intra-domain routing protection because eBGP routers do not have rich mesh-like connections with each other, and routes taken by different ASes are restricted by BGP policies. So, our previous intra-domain *e-cycle* construction algorithm, which is based on shortest path routing, cannot be directly applied for eBGP protection. In eBGP protection, we assume that AS_x considers protecting an eBGP link to AS_y only if there is at least a second link between these two ASes, directly or indirectly [5]. Normally, an AS should have at least two provider ASes to provide Internet connectivity [34], [35]. If any AS does not have parallel links to other ASes, it should have the incentives to negotiate with its provider ASes to build a backup eBGP link (see Section III-E). In this paper, we only consider protection of customer-provider eBGP links since failures of customer-provider eBGP links are the major cause of network connectivity problems. However, our methodology can be adapted to provide protection for peer-peer eBGP links if there exists parallel peer-peer eBGP links between two ASes.

Figure 6 shows a customer AS (call it AS_x) can have two types of connections to its provider ASes. One is that it is multi-connected by parallel links to the same AS (e.g., AS_y) as in Figure 6(a), and the other is that it is multi-connected to different ASes (e.g., AS_y and AS_z), as in Figure 6(b). We construct *e-cycles* for eBGP protection based on these two connection scenarios. Algorithm 2 shows how *e-cycle* protects an eBGP link l_i connecting AS_x and its provider AS_y. First, we try to find a parallel link connecting AS_x to the same provider AS_y (step 2-4). If no such parallel link can be found, then we need to choose an eBGP link connecting AS_x to a third party provider AS_z (step 6-14). Note that the chosen link l_j should have at least equal link capacity with l_i (step 7), and should not share the same Shared Risk Link Group (SRLG)[3] with l_i or any hidden AS between AS_x and AS_y (step 8). Otherwise, we need to select another eBGP link. After that, the major part of the cycle (the two links l_i and l_j and the traversed routers) of an *e-cycle* is determined (steps 18 and 20). Then, we need to choose different PTs in the *e-cycle* (steps 19 and 21).

[3]An SRLG is a group of links that are subject to a common risk, such as a link failure.

Algorithm 2 Inter-domain *E-cycle* Construction

//$AS\_speaker$(l_i): the BGP speaker pair of link l_i;
//$AS\_provider$(AS_x, l_i): the provider AS of AS_x at link l_i;
//$SRLG$(l_i, l_k): existence of a shared link between l_i and l_k;
//$HiddenAS$(l_i, l_k): existence of a shared hidden AS between l_i and l_k;
//$Router\_traversed$(R_x, R_y): the router set traversed from R_x to R_y;
Input: AS_x's inter-domain topology with neighboring ASes;
Output: $C = \{c | c = (V_c, S_c)\}$;

1: **for** (each eBGP link l_i) **do**
2: **if** (l_j is the parallel link connecting AS_y) **then**
3: AS_z = AS_y;
4: **continue**;
5: **else**
6: **while do**
7: find link l_j where link_capacity(l_j) $\geq$ link_capacity(l_i);
8: **if** (($SRLG$(l_i, l_j) == $\emptyset$) && ($HiddenAS$(l_i, l_j) == $\emptyset$)) **then**
9: **break**;
10: **else**
11: **continue**;
12: **end if**
13: AS_z = $AS\_provider$(AS_x, l_j);
14: **end while**
15: **end if**
16: [R_x, R_y] $\Leftarrow$ $AS\_speaker$(l_i);
17: [R'$_x$, R_z] $\Leftarrow$ $AS\_speaker$(l_j);
18: $insert\_cycle$(c, $router\_travsered$(R_x, R_z), $\emptyset$);
19: $update\_PT$(c, R_x, R_z);
20: $insert\_cycle$(c', $Router\_travsered$(R_y, R'$_x$), $\emptyset$);
21: $update\_PT$(c', R_y, R'$_x$);
22: **end for**

Different from intra-domain *e-cycle* construction, we can not select PT based on costs but need a specific rule: to protect an eBGP link l_i, both BGP speakers on l_i should act as PIs, and the PT for each of the PIs is the BGP speaker on link l_j in the other AS. Furthermore, we need to put the traversed routers between l_i and l_j in AS_x in the *e-cycle*. Since the provider AS (AS_y or AS_z) knows how to forward packets to AS_x, for easy deployment (e.g., to reduce the deployment complexity and protect ISP's privacy), we do not fully specify the sequence of routers in the cycle connecting l_i and l_j outside AS_x. Once the two PTs in *e-cycle* are chosen, we need to add two entries in the alternate forwarding table of routers in the cycle for identifying this *e-cycle*. Note that in *e-cycles* for eBGP protection, we only need to configure PTs for BGP speakers because other routers are protected by intra-domain protection. In this way, our proposed eBGP protection realizes eBGP protection with configuration involving at most three ASes.

Figure 6 shows an example of eBGP protection. If there are parallel links between two ASes which do not share SRLG and hidden AS, as shown in Figure 6(a), then we can directly build an *e-cycle*. For example, R1, R2, R3, R4 form an *e-cycle* to protect the eBGP link R2-R4, R2 and R4 act as the PIs, and R3 and R1 act as the PTs, respectively. It is simpler than the case when no parallel links can be found between two ASes and next we focus on analyzing this latter case. As Figure 6(b) shows, AS_y and AS_z are provider ASes of AS_x, and AS_y and AS_z may not be the neighbor AS

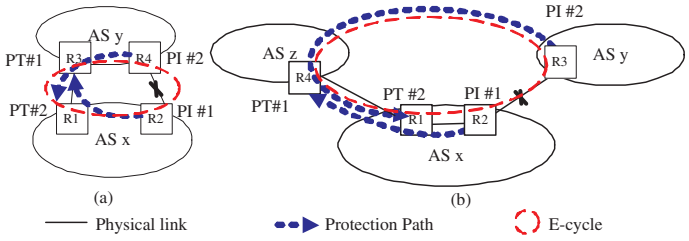


Fig. 6. *E-cycles* for link protection in inter-domain routing protection.

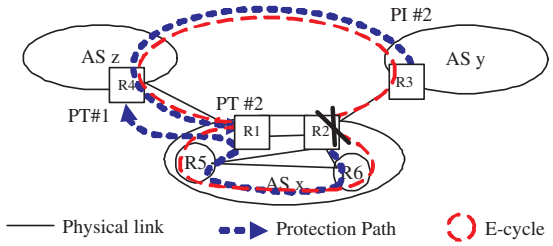


Fig. 7. *E-cycle* for node protection in inter-domain routing protection.

of each other. To protect link R2-R3, we select AS_z as the third provider AS, and we assume that link R1-R4 and R2-R3 do not share the same SRLG and any hidden AS, and the provider AS, AS_z, has a larger capacity to the Internet than the link load of R2-R3. Thus, we can build the protection path R2→R1→R4→⌊⌋→R3. Once link failure between R2 and R3 is detected, router R2, which acts as a PI, will activate the protection path to router R4 (PT), which will then take the responsibility to forward packets along normal routes. Similar to intra-domain routing protection, we need to protect reverse traffic over failed links. Remote ASes may not know the link failures immediately. For example, traffic whose destination is AS_x will reach AS_y eventually based on the routes learned from BGP but will fail to get to AS_x after link R2-R3 fails. At present, most solutions do not consider this problem [12]. Fortunately, our solution can solve this problem by activating the second protection path, R3→⌊⌋→R4→R1. In this context, R3, which acts as the second PI (with the corresponding PT R1), can launch the second protection path. In order to protect the eBGP link R1-R4, similar protection paths can be built.

Note that the *E-cycle* Construction Algorithm for Inter-domain routing does not require specifying all router information in an e-cycle. For example, as shown in Figure 6(b), for any upstream packets from AS_x or downstream packets to AS_x, the packets will be capsulated as *e-cycle* packets with specified *e-cycle* ID. The *e-cycle* ID will be identified only by AS_y and AS_z. Other other ASes between AS_y and AS_z (if they are not directly connected) do not need to identify the *e-cycle* ID, and the *e-cycle* packets can be forwarded by the IP header in these ASes. Thus, to deploy an inter-domain *e-cycle*, an AS only needs to take the neighbor ASes into account and evaluate if the eBGP links connecting the neighbor ASes can protect other eBGP links. ASes do not require the knowledge of traversed topology since the IP headers of *e-cycle* packets ensure successful packet delivery to PTs. Although the eBGP protection in this example above mainly tackles with the protection of customer-provider eBGP links, *e-cycle* also can provide protection for peer-peer eBGP links. if two ASes only connected by single peer-peer eBGP link, we cannot simply use other peer-peer eBGP links to provide protections, which will violate the BGP policy issue.

D. BGP(iBGP/eBGP) Node Protection

Sections III-B and III-C provide node and link protection in intra-domain routing, and link protection for inter-domain routing (iBGP and eBGP), respectively. However, protection for BGP nodes is not well addressed because there may be only one available egress point to destinations within an AS, and the protection paths built in Section III-B and III-C can not successfully provide failure recovery under BGP node failures. As shown in Figure 6, R2 is the only available egress point to AS_y, which means that all traffic to AS_y will be forwarded by R2. Although R1 has another routing path to AS_y, the traffic from R1 to AS_y will still get to AS_y via R2 because this routing path $\{AS_x, AS_y\}$ is chosen by BGP as the best path to AS_y.

Assuming that R2 fails, the protection path from R2 to R4 will be broken and then traffic from AS_x will be dropped though traffic from AS_y will get to AS_x. Fortunately, we can provide BGP node protection by extending our *e-cycle* construction for eBGP link protection (c.f., in Section III-C). That is, we can put the neighbors of the failed BGP node to extend *e-cycles* built for eBGP link protection. Figure 7 illustrates the example. Different from the *e-cycle* in eBGP link protection in Figure 6(a) where the *e-cycle* is R2→R1→R4→⌊⌋→R3, the *e-cycle* for BGP node protection includes all neighbors of R2 and is specified as R2→R6→R5→R1→R4→⌊⌋→R3 in which the PT of R2, R5 and R6 is R1. Then, if R2 fails, then any traffic from R1, R5, or R6 to AS_y will be encapsulated and forwarded to R4, and R4 will help them forward the traffic to AS_y eventually. However, if routing protection is not deployed in the network, the TCP connections and BGP session between R2 and R3 and between R1 and R2 will be broken after R2 fails, which may induce changes of the BGP routing tables and routing convergence processes. In particular, lots of packets may be dropped during routing convergence.

Note that *e-cycle* built for BGP node protection also provides protection for intra-domain routing nodes at the same time. As Figure 7 shows, if we build an *e-cycle* for R3 (i.e. R2-R6-R5-R1-R4-⌊⌋-R3) and link R2-R6 fails, then traffic from R2 to R6 can be forwarded through the protection path R2→R3→⌊⌋→R4→R1→R5→R6 and traffic from R6 to R2 can be forwarded through the protection path R5→R1→R4 →⌊⌋→R3→R2. For the purpose of deployment efficiency, we may not build extra *e-cycles* for R5 and R6. Then, this built protection path of our *e-cycle* approach is similar to what is being constructed for intra-domain routing protection in [17]. In future work, we will further study the joint optimization of the virtual cycle design for both intra- and inter-domain routing protection to minimize the total number of extra FIB entries required.

E. Discussion

Virtual Cycle Construction In *e-cycle*, virtual cycle construction is an important procedure for deployment in operational networks. Normally, each node under protection is covered by one virtual cycle. For intra-domain routing protection, we directly adopted the virtual cycle construction algorithms for *p-cycle* [29], [30] in Section III. Since routers have mesh-like connections between themselves, network operators can directly construct virtual cycles for them according to their preferences. Since *e-cycle* is designed for routing protection in IP networks, each *e-cycle* packet has an IP packet header. Thus, if the next-next neighbor is not included in the same cycle, *e-cycle* can use normal routes to forward packets to the right destinations according to the IP addresses of PTs. For inter-domain routing protection, virtual cycle construction is much easier than that in intra-domain routing protection, and a virtual cycle can be configured and built between eBGP speakers connected by two parallel eBGP links.

Computation Complexity Different types of routing protocols have different computation overheads of *e-cycle*. For inter-domain routing protection, *e-cycles* can be built between any two parallel eBGP links and BGP border routers connected by these eBGP links will be chosen as PI and PT, respectively. Thus, we do not introduce any computation overhead. However, for intra-domain protection, we need to run SPT computations to identify PT for each PI. The computation complexity is O(|E|+|V|log|V|) where |E| is the number of links and |V| is the number of nodes. The computation complexity is smaller than the existing IP-FRR proposals, such as Tunnels and Not-Via [15]. Moreover, *e-cycle* construction algorithms are off-line computation procedures, and the computation overheads will not introduce computations to routers.

Implementing *e-cycle* in Routers *E-cycle* requires changing the forwarding plane of routers. However, it does not need to change the router hardware. Normally, routers use field-programmable gate arrays (FPGA) or network processors in linecards to process the IP options including TTL such that we only need to re-configure the programmable logic of the hardware to realize e-cycle in routers. In *e-cycle*, we use Bidirectional Forwarding Detection (BFD) [36] to detect link failures. We only need to configure PT for each link failure, which can be realized by extending BFD implementations. To easily realize *e-cycle* in traditional routers, we can encode *e-cycle* labels in IP options. Thus, we can implement e-cycle in routers by re-configure the logic of FPGA or network processors in linecards such that routers can process e-cycle packets according to the e-cycle labels and TTLs in the IP options. We will provide an alternate to deploying e-cycles by setting up dedicated labels/tunnels in Section V, which does not require the modifications in any implementation of forwarding planes.

Real Deployment Agreements Most ASes have multi-homing eBGP links with their provider ASes [37], [34], [35], e.g., for the purpose of routing reliability and traffic engineering. Normally, an AS have at least two provider ASes to provide Internet connectivity [34], [35], and provider ASes will announce their network connectivity and provide transit service. Virtual cycles can be constructed between these eBGP links, and the protection paths will be activated between customer ASes and their provider ASes. If ASes involved in an *e-cycle* are competing, we can simply solve the issue by partially deploying *e-cycles* between the customer and its provider ASes since these ASes are not competing normally. For example, as shown in Figure 6, assuming the provider ASes, i.e., AS y and AS z, are competing, the customer AS, i.e., AS x, can negotiate them to build two *e-cycles* for protection of link R2-R3, respectively. AS x can negotiate with AS z to build an *e-cycle* to protect the traffic from R2 to R3 and the PT of the *e-cycle* is set to R4, and AS y does not need to get involved to this *e-cycle*. Similarly, AS x can build another *e-cycle* with AS y to protect reverse traffic from R3 to R2 and the PT of the *e-cycle* is set to R1, and this *e-cycle* does not need to involve AS z. Note that, if an AS has only one available upstream eBGP link, it should have the incentives to negotiate with its provider ASes to build a backup eBGP link to configure an *e-cycle* for failure recovery.

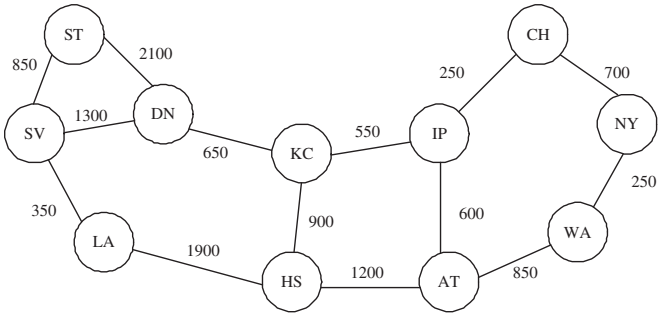
IV. Performance Evaluation

In this section, we evaluate the performance of our *e-cycle* approach by simulations. We firstly describe simulation setup in Section IV-A and then present the simulation results in Section IV-B.

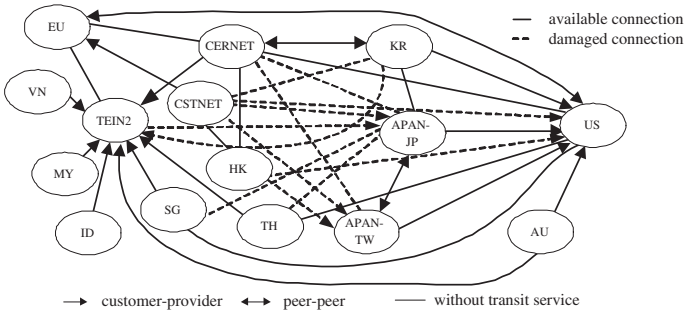
A. Simulation Setup

To evaluate our proposed solution, we implement a simulator that is able to simulate both intra- and inter-domain routing protocols [38]. In particular, the simulator considers BGP policy configurations, so that it can accurately evaluate the performance of eBGP protection. Our simulator simulates how a router would protect all end-to-end routing paths with different solutions including traditional IP-FRR [11] (including loop-free alternate (LFA), U-Turn Alternates (UTurns), Tunnels and Not-Via), Lightweight Not-Via [27], BGP-FRR [5], and R-BGP [12]. For each link in an end-to-end routing path, if no protection path is found, the simulator determines that the protection solution fails and can not provide protection for this failure. Since the *p-cycle* adoption proposed by Stamatelakis *et al.* [31] is similar to the original *p-cycle* solution [29] and they both can not be directly applied to IP networks, we only evaluate the performance of the original *p-cycle*. Normally, each node deployed with routing protection solutions activates a protection path to reroute the packets within several hundred microseconds after it detects the failure. However, the node will spend much more time in computing the rerouting path if the routing protection solution is not deployed in the node. For example, BGP requires several minutes to several tens of minutes or even more to compute a rerouting path. Thus, for simplicity, we do not present the performance of traditional routing fast convergence proposals in the paper.

Our evaluation uses real ISP topologies including the Abilene topology [39], the GEANT topology [40], and real ISP topologies provided by Rocketfuel [41], which we use to study intra-domain routing, and a simplified topology of the Asia-Pacific research networks [42], which we use to study inter-domain routing. Figure 8(a) shows the Abilene topology,



(a) The Abilene network



(b) Asia Pacific research networks

Fig. 8. Typical network topologies.

which is composed of 11 routers and 14 (28 directed) links. The intra-domain routing weight of each link is set according to the link delay [43]. GEANT and Rocketfuel topologies are omitted because of the space limitations, but can be found in [40] and [41], respectively. On the other hand, Figure 8(b) shows the Asia-Pacific research networks, which are composed of different ASes connected to EU and US [42]. The routing policies are obtained from APAN NOC [42] and CERNET NOC [44]. In Figure 8(b), dashed arrows in the network indicate failed links during Taiwan earthquake. Since we can not obtain detailed transit policies between different ASes in the Asia-Pacific networks, for simplicity, we only evaluate the performance with regard to reachability to US.

Routing protection solutions pre-compute rerouting paths which are activated when failures are detected. No convergence process is involved under failures in these solutions and thus traditional convergence performance can not effectively evaluate them. Although ***failure coverage*** was frequently used to evaluate routing protection performance in the literature [11], it does not consider whether failed links are indeed used for traffic forwarding. For example, there may exist some links which are not used for traffic forwarding (e.g., due to BGP policies and failures on these links) will not affect real traffic forwarding, and thus the metric can not accurately describe provisions of *end-to-end routing paths* under failures. We need to evaluate if a link is protected only when the link is used in the shortest path to a given destination. We consider an alternate definition of failure coverage that can capture more accurately the reachability of a node to the destination.

***Definition** 1:* **Valid failure coverage** *is the average success rate of routing protection for* every end-to-end routing path.

Note that valid failure coverage can measure uni- and bi-directional traffic between every end-to-end routing path.

Furthermore, we evaluate the overhead of existing solutions using the following metrics:

***Definition** 2:* **FIB Entry Increase Ratio** *is the ratio of the number of extra FIB entries for* all protection routing paths to *that of FIB entries required by traditional routing protocols*.

Normally, the number of intra-domain prefixes may not be so many [45]. For simplicity but without loss of generality, the used baseline is the number of FIB entries with traditional intra-domain routing protocols, i.e., OSPF, where we assume each router only has one prefix.

***Definition** 3:* **Path Inflation Ratio** *is the average ratio of increase in path length introduced by the protection paths.*

B. Simulation Results

Analysis of failure coverage. First, we study traffic forwarding in two directions between every end-to-end routing path in intra-domain routing. The *e-cycle* and *p-cycle* configuration in Abilene for all traffic in protection paths is shown in Table I, and configurations in GEANT and Rocketfuel networks are omitted due to the space limitations. Figure IV-B illustrates the *valid failure coverage* of intra-domain routing protection of different protection solutions in Abilene and GEANT. Since the lightweight Not-via achieves the same performance with Not-via, we do not plot it in the figure. Our simulation shows that our solution achieves 100% valid failure coverage for both uni- and bi-directional traffic. However, except Not-via, other existing solutions get a relatively low failure coverage, among which UTurns obtains the highest failure coverage of 84.48% and 68.97% for uni- and bi-directional protection, respectively. Thus, most of these solutions can not provide effective protection for failures. Note that we only evaluate if the connectivity between all source and end node pairs in the backbone networks is ensured with different routing protection schemes. As shown in Figure IV-B, we can observe similar results in GEANT topologies. *e-cycle* and Not-via achieve 100% failure coverage. Except these two schemes, LFA achieves the lowest failure coverage, and UTurns obtains the highest failure coverage. Moreover, we evaluate the failure coverage for bi-directional traffic in Rocketfuel topologies. As shown in Figure 9, Not-via and *e-cycle* always achieve 100% failure coverage in the 2-connected networks, and the failure coverage obtained by other approaches is round 80%.

We also evaluate the failure coverage of eBGP protection in the Asia-Pacific networks by simulating the failure caused by the Taiwan earthquake in 2006 [42]. Figure IV-B shows the results of different protection solutions. R-BGP achieves only about 45% failure coverage for both uni- and bi-direction traffic because no transit service is provided between EU and TEIN2 and the policy setting limits the effectiveness of protection. However, our solution well addresses this problem and provides 100% failure coverage with only two extra FIB entries. In the networks shown in Figure IV-B, the link between TEIN2 and EU and the link between EU and US are backbone links, and their capacity is much larger than the traffic load generated from their customer networks[4]. Thus, we can safely choose the BGP speaker in EU (i.e., GEANT)

[4]Since the information of *e-cycle* is intuitive, we do not give the details of the link capacity and the constructed *e-cycle*.

TABLE I
E-CYCLES FOR ABILENE

No.	*Traffic direction*	*Virtual cycles*	*Respective PTs in e-cycle*	*Decapsulation points in p-cycle*
1	Clockwise	ST-SV-LA-HS-AT-IP-KC-DN	[DN, AT, DN, SV, HS, KC, DN, HS]	[SV, LA, HS, AT, IP KC, DN, ST]
	Anticlockwise	ST-DN-KC-IP-AT-HS-LA-SV	[SV, KC, IP, LA, IP, IP, AT, AT]	[DN, KC, IP, AT, HS, LA, SV, ST]
2	Clockwise	KC-HS-AT-WA-NY-CH-IP	[NY, KC, KC, NY, AT, IP, KC]	[HS, AT, WA, NY, CH, IP, KC]
	Anticlockwise	KC-IP-CH-NY-WA-AT-HS	[IP, CH, HS, WA, CH, CH, AT]	[IP, CH, NY, WA, AT, HS, KC]

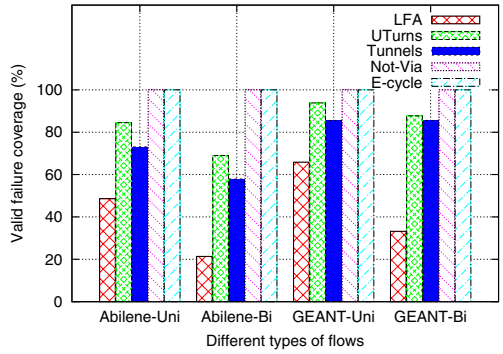


Fig. 9. Valid failure coverage of intra-domain routing protection.

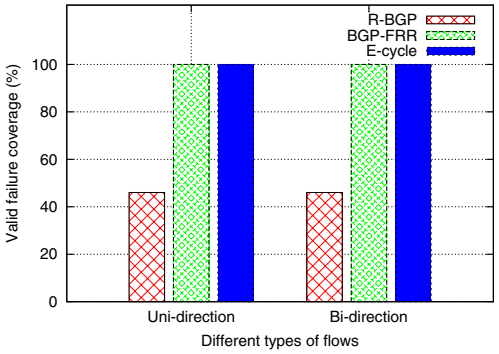


Fig. 11. Valid failure coverage of eBGP protection.

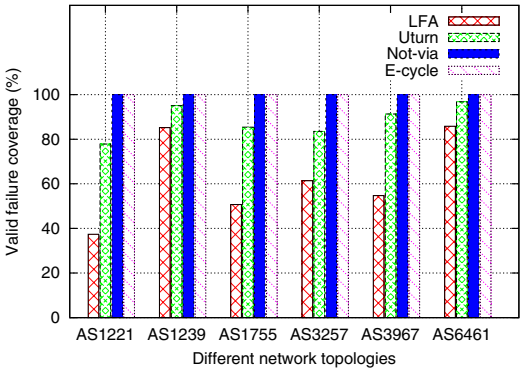


Fig. 10. Valid failure coverage for bi-directional traffic with Rocketfuel topologies.

as the PT to detour the failure detected by the PI in TEIN2 and forward traffic to US (i.e., Abilene), and then traffic from TEIN2 to APAN-JP and US will not be broken. Actually, GEANT started to provide transit service for CERNET after the earthquake by changing BGP policies in GEANT. However, our solution provides effective automatic traffic transit without operators' involvement. Moreover, protection will not result in link overload because the 10G link between EU and US provides enough capacity to carry the traffic impacted by the earthquake. Note that although BGP-FRR [5] can also provide 100% failure coverage, it only addresses failures in eBGP but not intra-domain routing (see Section II).

Analysis of overhead. We now evaluate the overhead introduced by different solutions. Here, we only analyze the intra-domain protection solutions which achieve 100% failure coverage, i.e., Not-Via, Lightweight Not-Via, *p-cycle*, and *e-cycle*. For simplicity, we only discuss one variant of Not-Via since we believe the cost of other variants, such as rNot-Via [19] and combination of LFA and Not-Via [20], are similar to Lightweight Not-Via and between that of Not-Via and *e-cycle*. We omit the results for inter-domain routing protection (e.g., BGP-FRR) as the results are also consistent. Figure IV-B shows the FIB entry increase ratio in Abilene and GEANT. Not-Via and Lightweight Not-Via require significantly more extra FIB entries, and increase the number of original FIB entries by 200% and 255%, respectively, in Abilene network. However, *p-cycle*, and *e-cycle* only requires only 27% extra FIB entries. Similarly, in GEANT network, the number of extra FIB entries in Not-Via and Lightweight Not-Via is increased by 200% and 322%. In Lightweight Not-Via, the extra FIB entries are constant, and the number is twice original FIB entries. Note that, since the extra FIB entries required by Not-via is proportional to the number of links, the number of extra FIB entries is increased in the increase of network size. Both *p-cycle*, and *e-cycle* only require 11.5% extra FIB entries.

We also evaluate the path inflation of these solutions. Figure 12 shows the path inflation results calculated according to the actual path length of packet forwarding due to the protection paths. It is not surprising that Not-Via and Lightweight Not-Via introduce around 20% of path inflation in Abilene and GEANT because they require that Not-via packets are firstly forwarded to the nodes on the opposite side of the nodes encapsulating the packets and then forwarded according to normal routes after packet decapsulation. *p-cycle* induces more than 32% path inflation since it uses longer detour routing paths than Not-Via. *e-cycle* effectively reduces path inflation, and the path inflation ratio in the Abilene and GEANT is about 21% and 13%, respectively.

Summary. In our experiments, we show that *e-cycle* provides 100% valid failure coverage for both intra- and inter-domain routing. While Not-via [15] and its variation [27] provide 100% failure coverage for intra-domain routing, they both require more than 100% of extra FIB entries. While *p-cycle* can achieve better failure coverage with less extra FIB

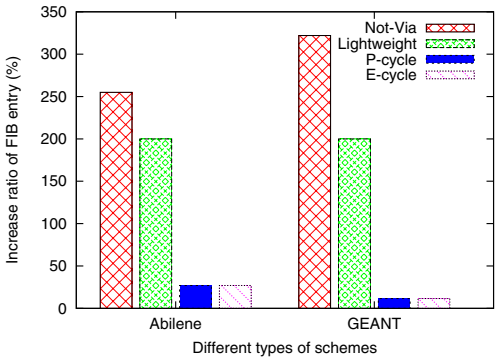


Fig. 12. Increase ratio of FIB entry in intra-domain routing protection.

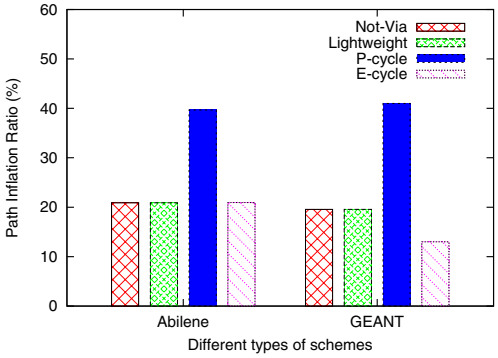


Fig. 13. The increase ratio of path length in intra-domain routing protection.

entries than other existing solution if it can be adopted in IP networks, it will introduce a higher path inflation ratio than *e-cycle*. In short, *e-cycle* provides a practical solution to routing protection.

V. Experimental Study in Real Networks

Section III-E presents an approach to realizing *e-cycle* by modifying data-plane implementations. However, it may not be easy to fully deploy *e-cycle* in real production networks with such modifications. In this section, we present an alternate to deploying e-cycles by setting up tunnels, and then study the performance of our path protection solution in operational networks to investigate the practicability of *e-cycle* and its incremental deployment. The goal of the experiment is to show the practicality and deployability of e-cycle using the existing forwarding technique. In particular, we explore the implications of routing protection, i.e., to investigate whether the packet forwarding performance will be exacerbated by packet fragmentation induced by routing protection, which are not addressed in the literature.

A. Methodology

In order to study the practicability of *e-cycle*, we deploy it in some operational networks of Chinese ISPs. As an instance of our *e-cycle* solution, we choose a label-based tunnel protocol, Layer 2 Tunneling Protocol (L2TP) [33], as the protection technique to realize *e-cycle*. L2TP is well supported in mainstream commercial routers. With L2TP, it is easy to deploy our solution in operational networks and provide incremental deployment of *e-cycle*. L2TP can forward encapsulated packets to detour routing failures using pre-configured directed forwarding, and hence form a tunneling path. In this context, L2TP Access Concentrators(LACs) [33] act as PIs and L2TP Network Services (LNSes) [33] act as PTs. In particular, L2TP provides reliable connections, we can easily infer whether PIs can get to PTs in this experiment.

Figure 14 shows the topology in our experiments. We deploy hosts at different ASes that belong to three different ISPs to study *e-cycle* for routing protection. The information of ASes is shown in Figure 14. One host (S1) is deployed in AS 4808 of CNC China as the source, and one host (D1) acting as the destination is deployed in AS 4538 of CERNET. We deploy the third host in AS 17964 acting as a PT. Here, we consider one specific path AS 4808 $\rightarrow$ AS 4837 $\rightarrow$ AS 4538, which we term the *normal path*. We observe that throughout our experiments, the path segment in AS 4837 generally experiences high congestion in forwarding packets along the normal path. One possibility is that AS 4837 is a backbone network and carries heavy traffic in general. Thus, we activate *e-cycle* to detour the normal path. A tunneling path between S1 in AS 4808 and the PT in AS 17964 will be activated, and we term the new routing path between S1 and D1 via the tunneling path to be the *protection path.* Note that, as the e-cycle we designed for eBGP protection (see Section III-C), we only need to specify PI and PT in e-cycle and we do not need to specify other router information in e-cycle for eBGP protection. We only evaluate the e-cycle to protect eBGP link S1-D1, because, from the point of view of protecting eBGP link S1-D1, the e-cycle is complete. Most routers that the protection path traversed are non-backbone networks, which carry less traffic in general. Based on the pre-configured protection path, we will measure the performance of the traffic with the normal path and the protection path, respectively.

To collect the detailed routing information, we use the traceroute tool to collect IP-level router information, and map the IP-level tracroutes to the corresponding AS-level traceroutes by mapping IP addresses to their origin ASes based on BGP routing tables. To investigate whether routing protection can achieve expected forwarding performance, we will measure and compare packet loss, round trip time (RTT) and throughput in different routing paths.

In our experiments, we only focus on inter-domain routing. Similar methodologies can be applied to the case of intra-domain routing.

B. Experimental Results

Experiment 1 (Packet loss): We randomly choose ten different times in five different days to generate 100 ping packets and compare the average packet loss rate of traffic. Figure 15 shows the packet loss rate of different flows with different routing paths. In Figure 15(a), we measure the loss rate of small packets with different paths. It shows that the protection path reduces the packet losses of the normal path. While the activation of the protection path (i.e., path switching) may trigger some packet loss, the loss rate remains

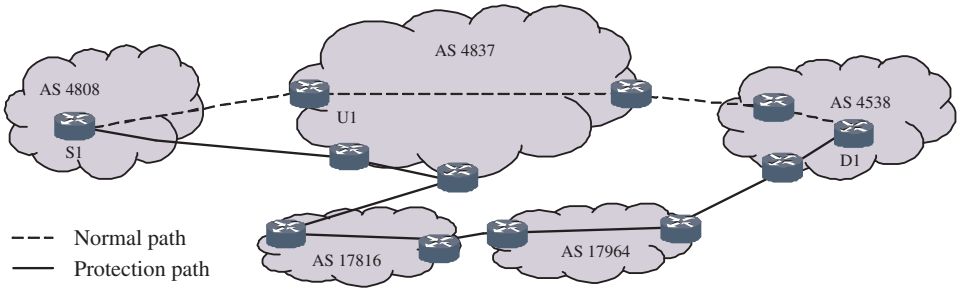


Fig. 14. BGP peer relationship examples in China.

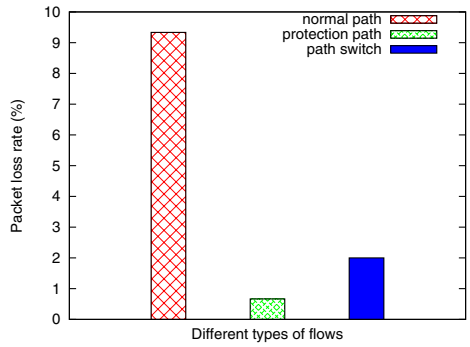
TABLE II
DIFFERENT MTU SIZE OF THE TRAFFIC FLOW AND PT

Routing Type	Traffic flow	LNS
Normal Path	1452bytes	1492bytes
Protection Path	1372bytes	1492bytes

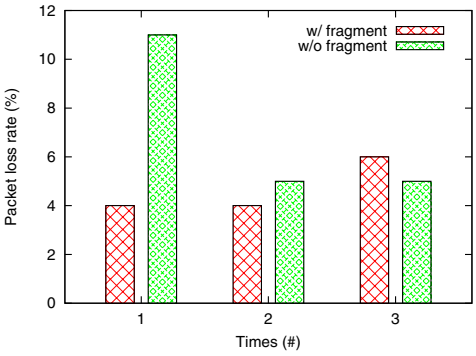
lower than that in the normal path. Overall, we observe that the use of the protection path can reduce the packet loss rate resulting from the normal path (which experiences congestion in our experiments). We will compare the performance in the normal path and the protection path under different situations with different experiments.

Furthermore, we also investigate the loss rate of large packets, and consider the case where fragmentation is introduced by tunneling. Table II shows the MTU of the traffic flow and PT. The MTU in the normal path is 1452 bytes and that in the protection path is 1372 bytes. We measure packet loss when packets are de-fragmented in protection path, and we generate the ping flow with different byte sizes, 1372 bytes and 1373 bytes, respectively. Figure 15(b) shows the average measurement results that we obtain during three different periods. We observe that the average packet loss rates with or without fragmentation are both about 6%. Thus, fragmentation introduced by tunneling does not exacerbate packet loss. Note that packet fragmentation in congested networks may exacerbate packet forwarding performance. However, protection paths in e-cycle that we selected will not induce congestions since we can carefully consider the link capacity during computing and constructing e-cycles (see Algorithm 2). Thus, we do not observe that fragmentation of e-cycle packets exacerbates packet forwarding performance.

Experiment 2 (Round Trip Time (RTT)): We also study the round trip time (RTT) in each path. We first analyze the router hops in different ASes for calculation of the transmission delay in every AS. Figure 16 shows RTT results for small packets. We observe that the average RTT for small packets in the whole protection path is about 3.8 ms, which is smaller than 19.78 ms in the normal path that experiences congestion. Moreover, we measure the RTT for large packets with/without fragmentation, and investigate whether fragmentation introduced by tunneling exacerbates transmission performance. Figure 17 illustrates the average RTTs in the protection path with or without fragmentation. We measure the results at three different times, and the average RTTs with and without fragmentation are 16.67 ms and 14 ms, respectively. Because of the same reason we discussed above, the protection path achieves better performance than the normal path. Thus, fragmentation by tunneling does not exacerbate the transmission delay in the protection path.



(a) Small packet loss in different paths



(b) Large packet loss in protection path

Fig. 15. Packet loss rates with different paths.

Experiment 3 (Throughput): We use *wget* to measure the throughput of 10 different times that span over five different days. Figure 18 shows the throughput results of both normal and protection paths in each test run. The protection path achieves much higher throughput than the normal path (187.68 MB/s vs. 15.58 MB/s, respectively). Note that the throughput in the protection path is measured under packet fragmentation, because the transferred data in *wget* is larger than the MTU shown in Table II.

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we propose a unified protection solution called *e-cycle* for intra- and inter-domain routing to efficiently

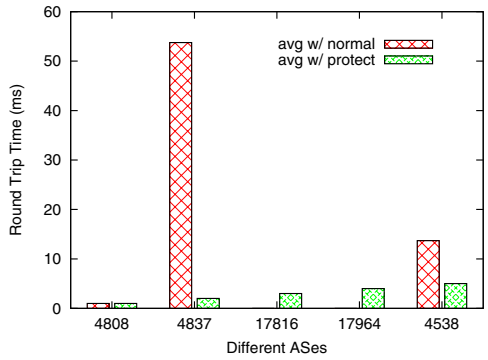


Fig. 16. Round trip time of the traffic.

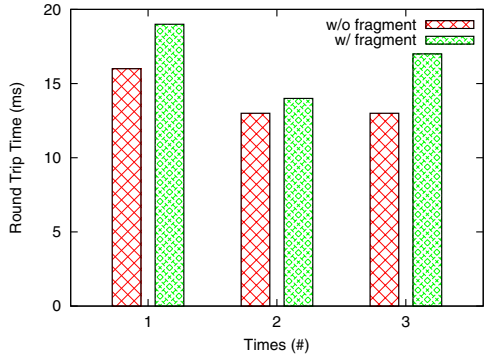


Fig. 17. RTTs with packet fragmentation.

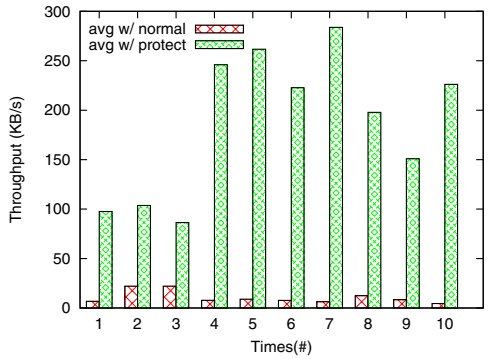


Fig. 18. The throughput of the flow with two different paths.

recover from routing failures. Specifically, *e-cycle* constructs protection paths and provides node and link protection. Simulation results show that our proposed solution achieves 100% failure coverage in both intra- and inter-domain routing. Moreover, we partially deployed our solution in operational networks to demonstrate the practicality of our solution. We find that our solution will not introduce much overhead to packet forwarding.

We are currently integrating the *e-cycle* solution into some commercial routers and will *fully* deploy them on CERNET and CERNET2 (the China Education and Research NETwork [44]) to study its real performance in larger-scale operational networks. In future work we are trying to propose an optimal virtual cycle design for *e-cycle* by applying Integer Liner Programming (ILP) algorithm to find minimized cycles without candidate cycle enumeration in intra-domain routing protection. We will further jointly optimize virtual cycle design for both intra- and inter-domain routing protection to minimize the total number of extra FIB entries required.

ACKNOWLEDGEMENT

We would like to thank Pierre François and David. K.Y. Yau for very helpful feedback and suggestions. We also thank Xiaofen Fang for helping us setup the experiment environment.

REFERENCES

[1] A. Markopoulou, G. Iannaccone, S. Bhattacharyya, C. Chuah, and C. Diot, "Characterization of failures in an IP backbone," in *Proc. 2004 IEEE INFOCOM*, pp. 2307–2317.
[2] C. Labovitz, A. Ahuja, A. Bose, and F. Jahanian, "Delayed Internet routing convergence," *IEEE/ACM Trans. Networking*, vol. 9, no. 3, pp. 293–306, 2001.
[3] Y. Afek, A. Bremler-Barr, and S. Schwarz, "Improved BGP convergence via ghost flushing," *IEEE J. Sel. Areas Commun.*, vol. 22, no. 10, pp. 1933–1948, 2004.
[4] D. Pei, M. Azuma, D. Massey, and L. Zhang, "BGP-RCN: improving BGP convergence through root cause notification," *Computer Network*, vol. 48, no. 2, pp. 175–194, 2005.
[5] O. Bonaventure, C. Filsfils, and P. Francois, "Achieving sub-50 milliseconds recovery upon BGP peering link failures," *IEEE/ACM Trans. Networking*, vol. 15, no. 5, pp. 1123–1135, 2007.
[6] M. Shand and S. Bryant, "IP fast reroute framework," RFC5714, Jan. 2010.
[7] A. Atlas, A. Zinin, R. Torvi, G. Choudhury, C. Martin, B. Imhoff, and D. Fedyk, "Basic specification for IP fast-reroute: loop-free alternates," RFC 5286, Sep. 2008.
[8] Q. Li, M. Xu, L. Pan, and Y. Cui, "A study of path protection in self-healing routing," in *Proc. 2008 IFIP Networking*, pp. 554–561.
[9] T. Cicic, A. F. Hansen, A. Kvalbein, M. Hartmann, R. Martin, M. Menth, S. Gjessing, and O. Lysne, "Relaxed multiple routing configurations: IP fast reroute for single and correlated failures," *IEEE Trans. Network and Service Management*, vol. 6, no. 1, pp. 1–14, 2009.
[10] K. W. Kwong, R. Guérin, A. Shaikh, and S. Tao, "Balancing performance, robustness and flexibility in routing systems," *IEEE Trans. Network and Service Management*, vol. 7, no. 3, pp. 186–199, 2010.
[11] P. Francois and O. Bonaventure, "An evaluation of IP-based fast reroute techniques," in *Proc. 2005 ACM CoNEXT*.
[12] N. Kushman, S. Kandula, D. Katabi, and B. Maggs, "R-BGP: staying connected in a connected world," in *Proc. 2007 NSDI*.
[13] L. Wang, M. Saranu, J. Cottlieb, and D. Pei, "Understanding BGP session failures in a large ISP," in *Proc. 2007 IEEE INFOCOM*.
[14] Q. Li, M. Xu, J. Wu, X. Shi, D. Chiu, and P. Lee, "Achieving unified protection for IP routing," in *Proc. 2010 IEEE ICCCN*.
[15] S. Bryant, M. Shands, and S. Previdi, "IP fast reroute using not-via addresses," Internet draft, draft-ietf-rtgwg-ipfrr-notvia-addresses-05.txt, Mar. 2010.
[16] S. Nelakuditi, S. Lee, Y. Yu, Z. Zhang, and C. Chuah, "Fast local rerouting for handling transient link failures," *IEEE/ACM Trans. Networking*, vol. 15, no. 2, pp. 359–372, 2007.
[17] H. Wang, Y. Yang, P. Liu, J. Wang, A. Gerber, and A. Greenberg, "Reliability as an interdomain service," in *Proc. 2007 ACM SIGCOMM*, pp. 229–240.
[18] Y. Wang, H. Wang, A. Mahimkar, R. Alimi, Y. Zhang, L. Qiu, and Y. R. Yang, "R3: resilient routing reconfiguration," in *Proc. 2010 ACM SIGCOMM*, pp. 291–302.
[19] A. Li, P. François, and X. Yang, "On improving the efficiency and manageability of NotVia," in *Proc. 2007 ACM CoNEXT*.
[20] M. Menth, M. Hartmann, R. Martin, T. Cicic, and A. Kvalbein, "Loop-free alternates and not-via addresses: a proper combination for IP fast reroute?" *Computer Networks*, vol. 54, no. 8, pp. 1300–1315, 2010.
[21] M. Hou, D. Wang, M. Xu, and J. Yang, "Selective protection: a cost-efficient backup scheme for link state routing," in *Proc. 2009 IEEE ICDCS*, pp. 68–75.
[22] Y. Yang, M. Xu, and Q. Li, "A light-weight IP fast reroute algorithm with tunneling," in *Proc. 2010 IEEE ICC*.
[23] M. Xu, Y. Yang, and Q. Li, "Selecting shorter alternate paths for tunnel-based IP fast reroute in linear time," *Computer Networks*, vol. 56, no. 2, pp. 845–857, 2012.

[24] Q. Li, M. Xu, Q. Li, D. Wang, and Y. Cui, "IP fast reroute: Notvia with early decapsulation," in *Proc. 2010 GLOBECOM*, pp. 1–6.

[25] K. Xi and H. J. Chao, "IP fast rerouting for single-link/node failure recovery," in *Proc. 2007 BROADNETS*, pp. 142–151.

[26] A. Kvalbein, A. F. Hansen, T. Cicic, S. Gjessing, and O. Lysne, "Multiple routing configurations for fast IP network recovery," *IEEE/ACM Trans. Networking*, vol. 17, no. 2, pp. 473–486, 2009.

[27] G. Enyedi, G. Rétvári, P. Szilágyi, and A. Császár, "IP fast reroute: lightweight Not-Via without additional addresses," in *Proc. 2009 IEEE INFOCOM*, pp. 2771–2775.

[28] P. François, O. Bonaventure, B. Decraene, and P.-A. Coste, "Avoiding disruptions during maintenance operations on BGP sessions," *IEEE Trans. Network and Service Management*, vol. 4, no. 3, pp. 1–11, 2007.

[29] W. Grover and D. Stamatelakis, "Cycle-oriented distributed preconfiguration: ring-like speed with mesh-like capacity for self-planning network restoration," in *Proc. 1998 IEEE ICC*, pp. 537–543.

[30] B. Wu, K. L. Yeung, and P. H. Ho, "ILP formulations for p-cycle design without candidate cycle enumeration," *IEEE/ACM Trans. Networking*, vol. 18, no. 1, pp. 284–295, 2010.

[31] D. Stamatelakis and W. Grover, "P-cycles: IP layer restoration and network planning based on virtual protection cycles," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 10, Oct. 2000.

[32] T. Cicic, A. Kvalbein, A. F. Hansen, and S. Gjessing, "Resilient routing layers and p-cycles: tradeoffs in network fault tolerance," in *Proc. 2005 HPSR*, pp. 278–282.

[33] J. Lau, W. Townsley, and I. Goyret, "Layer two tunneling protocol - version 3 (L2TPv3)," RFC 3931, Mar. 2005.

[34] J.-H. Park, P.-C. Cheng, S. Amante, D. Kim, D. McPherson, and L. Zhang, "BGP next-hop diversity: a comparative study," UCLA Computer Science Dept., Technical Report #100026, 2010.

[35] J. Choi, J.-H. Park, P.-C. Cheng, D. Kim, and L. Zhang, "Understanding BGP next-hop diversity," UCLA Computer Science Dept., Technical Report #100031, 2010.

[36] D. Katz and D. Ward, "Bidirectional forwarding detection," RFC 5880, June 2010.

[37] P. Mérindol, V. Van den Schrieck, B. Donnet, O. Bonaventure, and J.-J. Pansiot, "Quantifying ASES multiconnectivity using multicast information," in *Proc. 2009 IMC*, pp. 370–376.

[38] The routing protection simulator, http://qos.cs.tsinghua.edu.cn/~qli/software/simulator.zip.

[39] Abilene, http://www.internet2.edu/.

[40] GEANT, http://archive.geant.net/upload/pdf/GEANT-Topology-Map-220404_20040719132753.pdf.

[41] N. Spring, R. Mahajan, D. Wetherall, and T. Anderson, "Measuring ISP topologies with rocketfuel," *IEEE/ACM Trans. Netw.*, vol. 12, pp. 2–16, Feb. 2004.

[42] Y. Kitamura, Y. Lee, R. Sakiyama, and K. Okamura, "Experience with restoration of Asia Pacific network failures from Taiwan earthquake," *IEICE Trans. Commun.*, 2007.

[43] P. Francois, "Improving the convergence of IP routing protocols," Ph.D. dissertation, University Catholique de Louvain, Oct. 2007.

[44] "China education and research network (CERNET)," http://www.edu.cn/.

[45] C. Filsfils, P. Francois, M. Shand, B. Decraene, J. Uttaro, N. Leymann, and M. Horneffer, "LFA applicability in SP networks," Internet draft, draft-ietf-rtgwg-lfa-applicability-03, Sep. 2010.

Qi Li received the B.Sc. degree and the Ph.D. degree from Tsinghua University. His research interest includes network architecture and protocol design, system and network security.

Mingwei Xu received the B.Sc. degree and the Ph.D. degree from Tsinghua University. He is a full professor in Department of Computer Science at Tsinghua University. His research interest includes computer network architecture, high-speed router architecture and network security.

Jianping Wu received the Master and Ph.D. degrees in computer science from Tsinghua University, Beijing, China. He is now a full professor with the Department of Computer Science, Tsinghua University. He has published more than 200 technical papers in academic journals and proceedings of international conferences in the research areas of the network architecture, high-performance routing and switching, protocol testing, and formal methods.

Patrick P.C. Lee received the B.E. degree (first class honors) in Information Engineering from the Chinese University of Hong Kong in 2001, the M.Phil. degree in Computer Science and Engineering from the Chinese University of Hong Kong in 2003, and the Ph.D. degree in Computer Science from Columbia University in 2008. He is now an assistant professor of the Department of Computer Science and Engineering at the Chinese University of Hong Kong. His research interests are in network robustness and security.

Xingang Shi received the B.Sc. degree and the Ph.D. degree from Tsinghua University and the Chinese University of Hong Kong, respectively. His research interests include network measurement, streaming algorithms and routing protocols.

Dah Ming Chiu received the B.Sc. degree from Imperial College London and the Ph.D. degree from Harvard University in 1975 and 1980. After twenty years in industry (Bell Labs, DEC and Sun), he is currently the Department Chairman of Information Engineering with the Chinese University of Hong Kong (CUHK). His current research interests include P2P networks, network measurement, architecture and engineering, network economics, and wireless networks. He is an associate editor for IEEE/ACM TRANSACTIONS ON NETWORKING, and TPC member for various conferences including SIGCOMM, INFOCOM, and ICNP.

Yuan Yang received the Bachelor and the Master degree from Tsinghua University. He is now a Ph.D. student at Department of Computer Science in Tsinghua University. His major research interests include distributed routing protocol, computer network architecture and the next-generation Internet.