Achieving Unified Protection for IP Routing

Qi Li*, Mingwei Xu*, Jianping Wu*, Xingang Shi†, Dah Ming Chiu†, Patrick P.C. Lee‡
*Dept. of Computer Science, Tsinghua University, Tsinghua National Laboratory for Information Science and Technology
†Dept. of Information Engineering, ‡Dept. of Computer Science and Engineering, The Chinese University of Hong Kong
{liqi, xmw, jianping}@csnet1.cs.tsinghua.edu.cn, {sxg007, dmchiu}@ie.cuhk.edu.hk, pclee@cse.cuhk.edu.hk

Abstract—**Routing failures are common on the Internet and routing protocols can not always react fast enough to recover from them, which usually causes 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 intra- and inter-domain routing 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. 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.**

I. Introduction

Internet routing connecting different IP networks plays a critical role in ensuring packet delivery on the Internet. However, previous studies show that the current routing systems are not so effective in defending against failures. For instance, the 2006 earthquake in Taiwan caused global network disruption though unaffected links can provide potential connectivity for these networks. Prior work has shown that most failures are short term and last less than 3 minutes [13], such as Border Gateway Protocol (BGP) session resets and transient hardware failures. Unfortunately, current routing protocols fail to react quickly to recover from them, e.g., it is not unusual for BGP to take several minutes or longer to converge [10]. The significant recovery time leads to unreliable packet delivery.

Research has been done to investigate Internet routing convergence and fast reroute issues under routing failures. Fast routing convergence has been extensively studied in the literature [3], [14]. However, none of those solutions has been deployed in operational networks due to their complexity or in some cases subtle design flaws, e.g., Ghost Flushing [3] expedites convergence by sending extra route withdrawal but may exacerbate routing convergence in fail-over events. Basically, fast routing convergence is not effective for handling routing blackholes and loops. Especially, during short term failures [13], routing protocols do not usually have enough time to detect failures and start routing convergence.

To address these problems, another line of work is to realize routing protection by using backup routing paths [12], i.e., fast reroute approaches. IP-FRR solutions [15], [4], which are active subjects in the IETF, focus only on the protection of intra-domain routing. They share important drawbacks such as hard deployment and/or uncertain effectiveness, especially low protection effectiveness for bi-directional traffic over failed links. To address the fast reroute issue in inter-domain routing, Bonaventure *et al.* [5] proposed BGP fast reroute (BGP-FRR), the first solution to protect external BGP (eBGP) between different ASes, where manual configuration are required to eliminate shared links between eBGP speakers for effective automatic protection. R-BGP [9] was proposed to provide automatic effective failover for eBGP failures. However, R-BGP requires an extra Forwarding Information Base (FIB) entry for every prefix under protection, and the effectiveness is greatly restricted by routing policies. Neither BGP-FRR nor R-BGP considers internal BGP (iBGP) failures [17]. Moreover, they may greatly burden routers but provide uncertain effectiveness.

Previous studies consider protection only for single type of routing protocol and it is hard to deploy these solutions together to protect Internet routing in operational networks because of the deployment complexity. Thus, a light-weight unified routing protection solution is still lacking for real deployment. In this paper, we propose a unified based 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 intra- and inter-domain routing are highly correlated and a protection solution should not separate them, and by employing the unified solution we may greatly reduce the number of Forwarding Information Base (FIB) entries. In our solution, we propose enhanced protection cycle (*e-cycle*) to construct effective rerouting paths for node and link protection, and design different *e-cycle* construction algorithms for different routing protocols. Our solution eliminates extra route distribution and its protection effectiveness is not restricted by routing policies. Moreover, our solution is independent of specific routing protocols and is incrementally deployable. We evaluate our solution by simulations, and the results show that it provides 100% failure coverage in both intra- and inter-domain routing. We are currently integrating the *e-cycle* solution into some commercial routers and will deploy them on CERNET and CERNET2 (the China Education and Research NETwork [2]) to study its real performance in operational networks.

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 perfor-

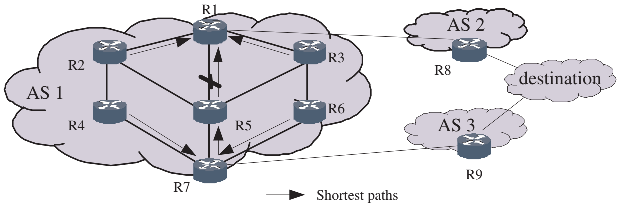


Fig. 1. Fast Reroute to network failure

mance of our solution in Section IV. We conclude this paper in Section V.

II. Problems in Existing Fast Reroute Solutions

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 [15], [6]. However, most solutions can not provide assured protection effectiveness and is not deployed in practice due to computational complexity. Among them, Not-via provides the best performance of failure coverage among various IP fast reroute (IP-FRR) solutions in intra-domain routing [6], [7]. The node protection scheme is recommended to detour failures and reduce the computational complexity [6], 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 will be activated to protect the link, 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, it is impossible to find a rerouting path to R1 not via R1 by node protection. The problem can be solved by applying the link protection scheme. That is, we can use not-via address to forward the packets **not via** link R1-R5. This would provide more effective protection at the cost of quadratic number of additional not-via addresses. Thus, it is obvious that it will introduce more overheads to compute and store extra FIB entries for all Not-via addresses.

Besides the above issues, intra-domain routing failure may trigger re-computation of inter-domain routing. For example, in the example above, If the protection for link R1-R5 fails, iBGP control messages between R1 and R7 will be dropped and the BGP session will be broken. Thus, all border routers in AS 1 will select AS 2 instead of AS 3 as the next hop to the destination, and all descendants ASes of AS 1 will recompute their routes to the destination. 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.* propose an automatic solution (BGP-FRR) specific for external BGP (eBGP) protection, and provide different protection strategies for different multi-homing stub networks [5]. Kushman *et al.* propose an improved BGP, R-BGP [9], 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 a FIB entry for every protected prefix in each router. These BGP protection solutions can not provide failure detour 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, quadratic number of extra FIB entries may be required.

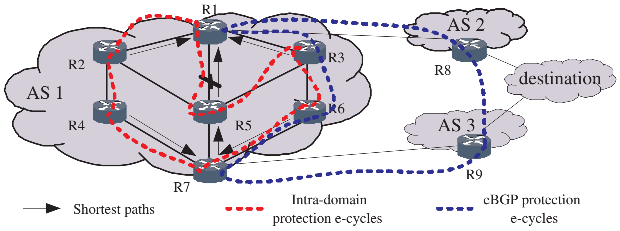


Fig. 2. Virtual cycles to recover from network failure

Virtual protection cycle (*p-cycle*) [16], [18] provides a practical and lightweight solution for routing protection, and it is firstly designed for failure recovery in SONET and WDM. In the *p-cycle* solution, minimal candidate virtual cycles are constructed to provide fast reroute for node and link failure recovery [18]. 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 [18]. Upon 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 advantages of *p-cycle* is that it provides protection for all nodes and links with a few extra FIB entries and flexibly handles multiple failures [16]. *P-cycle* 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 [6].

However, *p-cycle* has some drawbacks. The length of a rerouting path in the original *p-cycle* solution is greatly enlarged under failures, 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, different from nodes in WDM and SONET, 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 [16], their adoption is not a realistic one because it requires calculating the cost of every packet to destinations during packet forwarding. Furthermore, *p-cycle* 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 operated by different ISPs.

III. E-cycle: Enhanced Protection Cycle for Routing Protection

A. Overview of E-cycle

Different from traditional auto-discovery protection solutions which introduce much more complexity in routing protocols [5], [6], [9], enhanced protection cycle (*e-cycle*) provides efficient pre-configured routing paths to realize fast rerouting. Similar to *p-cycle* [16], *e-cycle* leverages virtual cycles to construct rerouting paths and uses different identifiers (*e-cycle IDs*) to identify these paths, thus provides protection for all nodes and links. The 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 tries 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 who detect failures and then activate protection paths to forward packets, and protection terminators (PTs) are routers who terminate protection paths and continue normal packet forwarding. Once an *e-cycle* is constructed, we need to select a PT for every PI in the cycle[1] and *e-cycle ID* based forwarding is only applied along the partial cycle between PI and PT.

When PI detects a failure, it starts to forward affected packets along the *e-cycle* towards PT. Since we want to bring as little overhead as possible to routers, the *e-cycle* ID is used as a label for direct forwarding. PI can simply implement this by encapsulating packets with a new IP header using IP encapsulation (e.g., L2TP [11]) to keep backward compatibility. The new packet header contains an *e-cycle* ID field which specify the unique identifier of the *e-cycle* used for fast forwarding, and a hop count field which specifies the hop count between PI and PT. The hop count indicates the lifetime of the packet in the *e-cycle*, and will decrease by one when that packet is forwarded by a router. If the hop count equals to zero, the packet will be removed from the *e-cycle* by PT and the original packet will be forwarded along a normal route to its destination.

Considering the same failure example in Figure 1 and assuming R5 as 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, *e-cycle* does not require all nodes in the cycle to understand *e-cycle* and to be configured with the same *e-cycle ID*. As shown in Figure 1, assuming 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.

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. *E-cycle* can handle all these conditions by detouring packets to PT as long as an *e-cycle* is pre-configured on PI. Thus, *e-cycle* provides much better efficiency by realizing a unified protection for both node and link failures. Since the construction algorithm, especially the selection of PT, will play a critical role in forwarding packets to the original destinations successfully, and intra- and inter-domain routing protocols have different forwarding features, we should construct different *e-cycles* for them. In the following subsections, we will discuss how to construct *e-cycles* to protect different types of routing protocols.

B. Intra-Domain Routing/iBGP Protection

Since iBGP relies on intra-domain routing, if we can guarantee intra-domain routing protection, iBGP link failures can be eliminated.[2] So next we only discuss *e-cycle* construction for intra-domain routing. Virtual cycle construction in IP networks is well studied in the literature [16], [18], and then we use the same construction algorithm as p-cycle to construct virtual cycles and focus on PT selection in these cycle paths here. The outline of PT selection is shown in Algorithm 1, which, based on the intra-domain topology, returns a set C, and each member c of 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 [16], [18](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 starting from R_i^c to traverse along the cycle, R_{i+1}^c is the next node to be encountered and R_{i-1}^c is the last one. If in the shortest path tree (SPT) rooted at R_i^c, $SPT\_Desc(\mathrm{R}_i^c)$ returns the descendants of the subtree under the failed link (or failed node), 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 use R_x^c to detour the failed link and forward packets to its descendent routers, R_x^c will never send them 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, 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 PT (step 17-19).

Figure 3 shows an intra-domain topology where the link weights are all set to 10, except that the weight of R5-R3 is 11. We assume that two virtual cycles indicated by the doted cycles are already constructed for these routers, and we need to choose a PT for each router in each cycle. For example, in Figure 3, 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 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

[1] In eBGP protection, not every node is required to act as PI. We will explain it in more details in a later subsection.

[2] 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 a separated paper.

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: $C = \{c|c = (V_c, S_c)\}$;
construct virtual cycles $C$=$\{c|c = (V_c, \emptyset)\}$;
for each $c \in C$ do
    for each $R_i^c \in V_c$ do
        flag = true;
        for ($R_x^c$ in [$R_{i+1}^c$, $R_{i+2}^c$, ..., $R_m^c$, $R_1^c$..., $R_{i-1}^c$]) do
            for each $R_y$ in $SPT\_Desc$($R_i^c$) do
                if ($R_i^c \in SPT\_traversed$($R_x^c$, $R_y$)) then
                    flag = false;
                    break;
                end if
            end for
            if (flag == true) then
                $update\_PT$($c$, $R_i^c$, $R_x^c$);
                break;
            end if
        end for
        if (flag == false) then
            $update\_PT$($c$, $R_i^c$, $R_{i+1}^c$);
        end if
    end for
end for
```

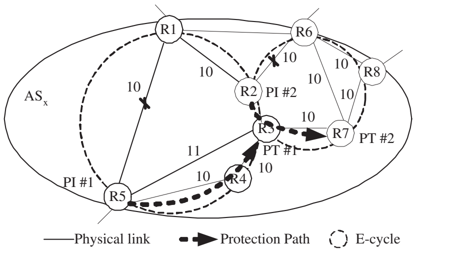
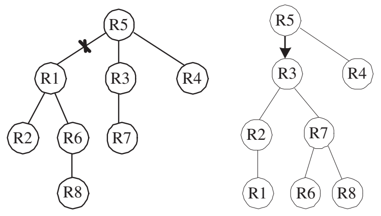


Fig. 3. *E-cycles* in intra-domain routing protection

as R5's PT since the shortest path from R4 to R1 is R4-R5-R1. For illustration, the shortest path tree rooted at R5 before and after the failure of link R1-R5 are depicted in Figure 4(a) and Figure 4(b), respectively. Once a PT is chosen, we need to distribute the alternative forwarding entries to the routers on this cycle for identifying this *e-cycle* address[3]. Figure 3 shows that we construct two *e-cycles* for eight routers in the AS. If any router detects a failure, it can launch the protection with a specific PT in the cycle. For example, in Figure 3, we assume that links R1-R5 and R2-R6 fail. 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.

C. eBGP Protection



(a) before failure (b) after failure

Fig. 4. The change of shortest path tree before/after network failure

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, *e-cycle* construction proposed for intra-domain routing protection, which is based on shortest path, can not be directly applied for eBGP protection. Figure 5 shows an customer AS, AS$_x$, can have two types of connections to its provider ASes [4]. One is that it is multi-connected by parallel links to the same AS, AS$_y$, as in Figure 5(a), and the other is that it is multi-connected to different ASes , AS$_y$ and AS$_z$, as in Figure 5(b). We construct *e-cycles* for eBGP protection based on these two connection features.

To effectively protect an eBGP link l_i connecting AS$_x$ and its provider AS$_y$, the *e-cycle* algorithm needs to consider different cases. First, we try to find a parallel link connecting AS$_x$ to the same provider AS$_y$. If no such a parallel link can be found, we need to choose an eBGP link connecting AS$_x$ to a third party provider AS$_z$. Note that the chosen link l_j should not share the same Shared Risk Link Group (SRLG) with l_i or any hidden AS between AS$_x$ and AS$_y$, and should have a larger link capacity than the load of l_i. Otherwise, we need to select another eBGP link. After that, the major part (the two links l_i and l_j) of an *e-cycle* is determined. Then, we need to choose different PTs in the *e-cycle*.

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 PI, and the PT for such a PI is the BGP speaker on link l_j in the other AS. Further, 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 packet 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. The algorithm is simple, and we do not present it here due to the page limitations.

Figure 5 shows an example of eBGP protection. If there are parallel links between two ASes which do not share SGLR and

[3] The entry distribution can be performed by manual configuration or an automatic mechanism, such as Label distribution protocol (LDP) as in Notvia [6].

[4] In eBGP protection, we assume that AS$_x$ considers to protect an eBGP link to AS$_y$ only if there is at least a second link between these two ASes, directly or indirectly [5].

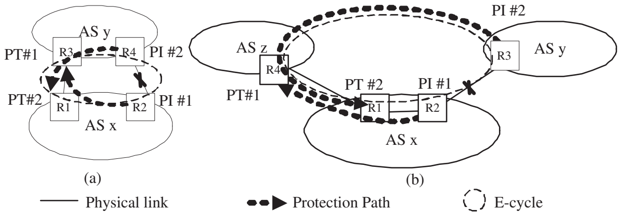
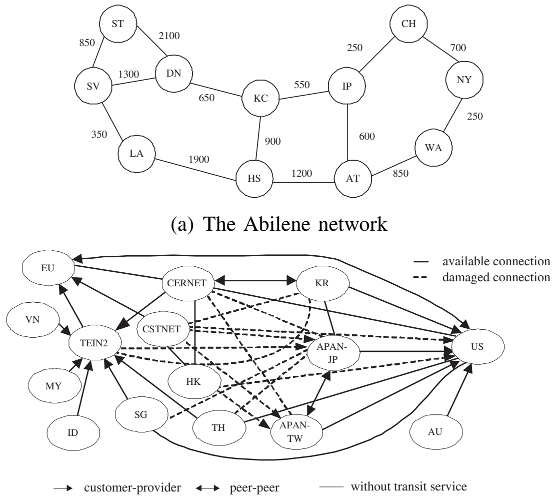


Fig. 5. *E-cycles* in inter-domain routing protection

hidden AS, as shown in Figure 5(a), we can directly build *e-cycle*. For example, R1, R2, R3, R4 form an *e-cycle* to protect the eBGP link R2-R4, R2 and R4 act as PI, and R3 and R1 act as PT, 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 5(b) shows, to protect link R2-R3, we select AS$_z$ as the third provider AS, and we assume that link R4-R1 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 acting as PI activates the protection path to router R4 (PT). PT will take responsibility to forward packets by normal routes. Similar to intra-domain routing protection, we need to protect reverse traffic over failed links. Remote ASes may not know the link failure 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 [9]. Fortunately, our solution solves this problem by activating the second protection path, R3→⎵→R4→R1. In this context, R3 acting 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.

IV. Performance Evaluation

To evaluate our proposed solution, we implemented a simulator that is able to simulate both intra- and inter-domain routing protocols. Especially, the simulator considers BGP policy, 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 [7] (including loop-free alternate (LFA), U-Turn Alternates (UTurns), Tunnels and Not-Via), BGP-FRR [5], and RBGP [9]. 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. To evaluate protection performance with different routing protocols, we use some real ISP topologies, including the Abilene topology [1] and a simplified topology of the Asia-Pacific research networks [8]. The Abilene topology, as shown in Figure 6(a), 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. The Asia-Pacific research networks are composed of different ASes connected to EU and US [8], as shown in Figure 6(b). The routing policies are obtained from APAN NOC [8] and CERNET NOC [2]. 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 EU, US, APAN and TEIN2.



(a) The Abilene network

(b) Asia Pacific research networks

Fig. 6. Typical network topologies

Although *failure coverage* was frequently used to evaluate routing protection performance in the literature [7], 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 due to BGP policies, and then the metric can not accurately describe provisions of *end-to-end routing paths* under failures because failures on these link will not impact real traffic forwarding. We consider an alternate definition that more accurately captures the reachability of a node to the destination as follows:

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.

First, we study traffic forwarding in two directions between every end-to-end routing path in intra-domain routing. The *e-cycle* configuration for clockwise traffic in Abilene is shown in Table I. Figure 7 illustrates the *valid failure coverage* of intra-domain routing protection of different protection solutions. Our simulation shows that our solution achieves 100% valid failure coverage for both uni- and bi-directional traffic. However, except Not-via, most 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. Although Not-via achieves 100% failure coverage, it requires dozens of extra FIB entries, while our solution only requires at most four extra entries. Thus, these solutions can not provide effective and efficient protection for failures.

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 [8]. Figure 8 shows the results of different protection solutions. R-BGP achieves only 5% failure coverage for both uni- and bi-direction traffic because

E-cycles	PT setting
ST-DN-KC-IP-AT-HS-LA-SV	[DN, HS, DN, KC, HS, SV, DN, AT]
KC-IP-CH-NY-WA-AT-HS	[NY, KC, IP, CH, NY, KC, KC]

TABLE I
E-CYCLES FOR ABILENE

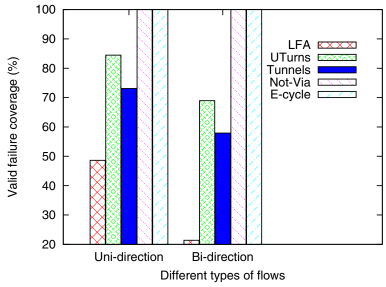


Fig. 7. Valid failure coverage of intra-domain routing protection

only the link CERNET-KR can provide potential protection for the failed link TEIN2-KR. Other failed links can not be effectively protected because no transit service is provided between TEIN2 and APAN-JP. However, our solution well addresses this problem and provides 100% failure coverage with only two extra FIB entries. In the networks shown in Figure 8, 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[5]. Thus, we can safely choose the BGP speaker in EU (i.e., GEANT) 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, out 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. Although BGP-FRR [5] can also provide 100% failure coverage, it does not consider link capacity and may introduce heavy traffic congestions which possibly induce more eBGP session failures.

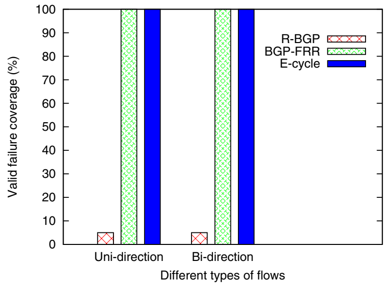


Fig. 8. Valid failure coverage of eBGP protection

V. CONCLUSION

In this paper, we have proposed a unified protection solution for intra- and inter-domain routing to efficiently recover from failures. Especially, *e-cycle* is proposed to construct protection paths and provide node and link protection. The simulation results show that our proposed solution achieves 100% failure coverage in both intra- and inter-domain routing. Currently, we are investigating and evaluating our solution under partial deployment in real operational networks, such as CERNET and CERNET2. 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. This research was supported in part by the 973 program (No. 2009CB320502), the 863 program (No. 2007AA01Z2a2 and No.2009AA01Z251), and the projects of National Science & Technology Pillar Program (No.2008BAH37B03).

REFERENCES

[1] Abilene. http://www.internet2.edu/.

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

[3] Y. Afek, A. Bremler-Barr, and S. Schwarz. Improved BGP convergence via ghost flushing. *IEEE Journal On Selected Areas In Communications*, 22(10):1933–1948, 2004.

[4] A. Atlas, A. Zinin, R. Torvi, G. Choudhury, C. Martin, B. Imhoff, and D. Fedyk. Basic specification for IP fast-reroute: Loop-free alternates. *Internet draft, draft-ietf-rtgwg-ipfrr-spec-base-10.txt*, November, 2007.

[5] O. Bonaventure, C. Filsfils, and P. Francois. Achieving sub-50 milliseconds recovery upon bgp peering link failures. *IEEE/ACM Transactions on Networking*, 15(5):1123–1135, 2007.

[6] S. Bryant, M. Shands, and S. Previdi. IP fast reroute using not-via addresses. *Internet draft, draft-ietf-rtgwg-ipfrr-notvia-addresses-01.txt*, July, 2007.

[7] P. Francois and O. Bonaventure. An evaluation of IP-based fast reroute techniques. In *Proceedings of ACM CoNEXT*, 2005.

[8] Y. Kitamura, Y. Lee, R. Sakiyama, and K. Okamura. Experience with restoration of asia pacific network failures from taiwan earthquake. *IEICE Transactions on Communications*, 2007.

[9] N. Kushman, S. Kandula, D. Katabi, and B. Maggs. R-BGP: Staying connected in a connected world. In *Proceeding of NSDI*, 2007.

[10] C. Labovitz, A. Ahuja, A. Bose, and F. Jahanian. Delayed internet routing convergence. *IEEE/ACM Transactions on Networking*, (3):293–306, 2001.

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

[12] Q. Li, M. Xu, L. Pan, and Y. Cui. A study of path protection in self-healing routing. In *Proceeding of IFIP Networking*, pages 554–561, 2008.

[13] A. Markopoulou, G. Iannaccone, S. Bhattacharyya, C. Chuah, and C. Diot. Characterization of failures in an IP backbone. In *Proceeding of the IEEE INFOCOM*, pages 2307– 2317, 2004.

[14] D. Pei, M. Azuma, D. Massey, and L. Zhang. BGP-RCN: Improving BGP convergence through root cause notification. *Computer Network*, 48(2):175–194, 2005.

[15] M. Shand and S. Bryant. IP fast reroute framework. *Internet draft, draft-ietf-rtgwg-ipfrr-framework-07.txt*, June, 2007.

[16] D. Stamatelakis and W. Grover. P-cycles: IP layer restoration and network planning based on virtual protection cycles. *IEEE Journal on Selected Areas in Communications*, 18(10), Octerber, 2000.

[17] L. Wang, M. Saranu, J. Cottlieb, and D. Pei. Understanding BGP session failures in a large ISP. In *Proceeding of the IEEE INFOCOM*, 2007.

[18] B. Wu, K. L. Yeung, and P. H. Ho. ILP formulations for p-cycle design without candidate cycle enumeration. *IEEE/ACM Transactions on Networking*, to appear.

[5] Due to the page limitations, we do not give the details of the link capacity and the constructed *e-cycle*.