A Backup Route Aware Routing Protocol – Fast Recovery from Transient Routing Failures

Feng Wang
School of Engineering and Computational Sciences
Liberty University
Lynchburg, VA 24502
fwang@liberty.edu

Lixin Gao
Department of Electrical and Computer Engineering,
University of Massachusetts, Amherst
Amherst, MA 01002
lgao@ecs.umass.edu

***Abstract*— As the Internet becomes the critical information infrastructure for both personal and business applications, survivable routing protocols need to be designed that maintain the performance of those services in the presence of failures. Routing failures that BGP can experience are failures that temporarily lose reachability during route convergence, also referred to as transient routing failures. This paper examines the survivability of interdoamin routing protocols in the presence of routing failure events, and provides a backup route aware routing protocol that performs non-stop routing in the presence of failures. We demonstrate through simulation its effectiveness in preventing packet losses during transient routing failures.**

I. Introduction

As the Internet starts to carry more and more mission critical services such as Voice-over-IP (VoIP) applications and games, there is a growing demand for the Internet to provide reliable services. Unfortunately, failures are fairly common in the Internet due to various causes such as maintenance, router crash, fiber cuts, and misconfiguration [5], [17]. When such failures occur, routing protocols should be able to quickly find alternate paths to provide forwarding continuity. Nevertheless, widespread routing failures in the Internet have been observed in experimental studies [14], [15]. Border Gateway Protocol (BGP), a path vector protocol for interdomain routing in the Internet, experiences considerable delay in reaching convergence [12], [13]. Previous studies have shown that end-to-end path performance degrades during routing convergence [2], [7], [13], [14], [17], [24], [28]. Furthermore, during several failure events, such as failover and recovery events, in which the reachability of the destinations is not compromised, BGP exhibits short-term routing table inconsistencies caused by the asynchronous route computation. These inconsistencies may cause short-term failures or routing loops [9], [22], [27], [28]. We refer to this transient loss of reachability as *transient routing failures*.

A basic strategy of building a resilient network is to use multi-path routing. A great effort has been taken to develop fast IP reroute scheme based on multi-path routing mechanism in the context of link state protocols [11], [16], [18]. In contrast to link state protocols, BGP is a path vector protocol where each router announces only its best path to its neighbors, which presents challenges in implementing multi-path routing for BGP. Existing multi-path routing approach focuses on the solutions for offering routing flexibility [29]. Unfortunately, this approach is not guaranteed to avoid transient routing failures.

Techniques were proposed [4], [10] to prevent the occurrence of transient routing failures. Bonaventure et al [4] propose a solution using pre-established protection tunnels to reroute traffic during failures. This approach is appropriate for resolving the problem of transient routing failures occurring within an AS. However, preventing transient routing failures across several ASes is more challenging and expensive. A more recent method, R-BGP protects data forwarding from failover events by providing recovery paths [10]. The limitation of the protocol is that it only focuses on providing fast recovery from failover on AS level. It does not provide fast recovery for iBGP and does not consider auxiliary failure events associated with recovery events. Our approach, a Backup Route Aware Routing Protocol (BRAP) is a full-fledged protocol that can provide survivable interdomain routing.

Our major contributions are summarized as follows.

- In contrast to existing approach [10], we develop a protocol that integrates several techniques to achieve fast transient failure recovery from various failure events occurring in the Internet, including failover and recovery events. One important feature of our approach is that a router always has at least one alternate path in addition to the best path, which are used to forward packets upon a failure.
- We discuss the practicality of deploying BRAP in a typical BGP system, where routing policies conforming to commercial agreements between ASes and the hierarchical iBGP configurations are deployed.
- We evaluate the effectiveness of our approach using measurement data from RouteView [20] and simulation. The results confirm that our approach can help ISPs to achieve non-stop interdomain routing and fast failure recovery.

The remainder of the paper is structured as follows. In Section II we describe traffic disruption due to various transient routing failures. In Section III, we describe BRAP in more detail. In Section IV, we show how to deploy BRAP in a typical BGP system. In Section V, we evaluate the effectiveness and efficiency of BRAP. Section VI presents related works, and we conclude in Section VII.

II. Packet Losses during Transient Routing Failures

In this section, we first define terminology through examples. And then, we illustrate how traffic disruption can happen during various failure events, such as network failure and repair events. In those two kinds of events, the destination is always physically connected with the network.

A. Definitions

BGP is a path vector protocol, in which it maintains routing information learned from neighbors. Each BGP router advertises the best route to its neighbors. If a router v's best path toward a destination is via router u, u is said to be the *primary neighbor* or *primary router* of v, while v is defined as the *upstream neighbor* of u. A router u is said to be the *backup neighbor* of router v if u is not the primary neighbor of v and u's best path to the same destination does not go through v. For example, in Fig. 1, router 3 is the primary neighbor of router 2 and router 1. Routers 1 and 2 are the upstream neighbors of route 3. Routers 1 and 2 are the backup neighbor of each other. Note that the local preference ranking for those paths in Fig. 1 is indicated with the order of the paths.

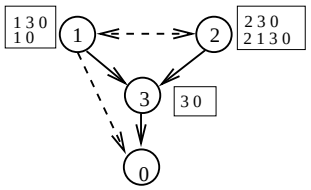

Fig. 1. Example used for illustrating concepts throughout the paper. The box around a node represents its routing table. The order of the paths indicates the local preference ranking for those paths. The solid arrow line represents a link in the best path, while the dashed line denotes a link in an alternative path.

If a router u is v's primary neighbor with respect to a destination, and v has a path to the destination not via u, the path is defined as a *reverse path* for router u.

Suppose that router u is router v's primary neighbor with respect to a destination. If router v has a path to the destination not via router u, the path is defined as a *reverse path* for router u. A upstream neighbor u of router v is called a *reverse neighbor* if router v receives the reverse path toward a destination advertised by router u. For example, in Fig. 1, router 1 has an alternate path not via its primary neighbor 3 so that the alternate path (1 0) at router 1 is a reverse path and router 1 is the reverse neighbor of router 3.

B. Packet Losses During Failover Events

By representing the path at the AS level, BGP hides the details of the topology and routing information inside each network. As a result, each BGP router does not know the complete, exact topology of the network. All it knows is the routes learned from its neighbors. When a router advertises its best route to neighbors, it cannot send the route back to its primary neighbor. In this case, the router will send a withdrawal message to *poison* its previous route in the neighbor, which is called *split horizon with poison reverse*.

We use an example shown in Fig. 1 to demonstrate packet loss during a failover event. Router 1 and 2 consider router 3 as next hop to access the destination. Since router 3 is the primary neighbor of router 1 and 2, router 1 and 2 are not allowed to advertise their best paths back to 3. Thus, router 3 does not realize the existence of the reverse path (1 0) at 1. Now suppose that the link between 3 and the destination fails. In this case, router 3 loses connection to the destination. In order to find other available paths, router 3 has to send a withdrawal message to activate 1 to select the reverse path. Before router 3 receives the new path from 1, it temporarily loses its connection to the destination. The back and forth procedure can last from several seconds to tens of seconds [27], [28].

C. Packet Loss During Recovery Events

We consider another example shown in Fig. 2 to demonstrate how transient failures can occur during a link repair event. In BGP, route information exchanged between routers is policy-conformant routes. Routing policies are set locally according to business agreements [8]. The implication of policy-based routes is that not every two routers that have a physical path connecting them can indeed exchange information. That is, connectivity does not necessarily mean reachability.

In Fig. 2(a), router 1 uses the direct path (1 0) as its best route, and router 2 and 3 use the paths (2 1 0) and (3 1 0) as their best paths, respectively. Suppose that once the link between router 3 and 0 is recovered, router 3 switches to the direct path (3 0), and then propagates the path to 1 and 2. Suppose that router 1 will switch to the recovery path (1 3 0) as shown in Fig. 2(b). Suppose that according to their routing policy, router 1, 2 and 3 cannot propagate the path learned from one router to another. As a result, router 1 has to send a withdrawal message to 2 to withdraw its previous announced route. If the withdrawal message arrives at 2 earlier than the recovery path (2 3 0) sent by 3, router 2 will temporarily lose its connection to the destination.

In addition, packet loss caused by a link recovery event can occur among iBGP routers. The temporary loss of reachability to a destination is due to the iBGP constraint, i.e., a route received from an iBGP

(a) Before a recovery event (b) After a recovery event

Fig. 2. A transient failure experienced by router 2 during the link between router 0 and router 3 is recovered.

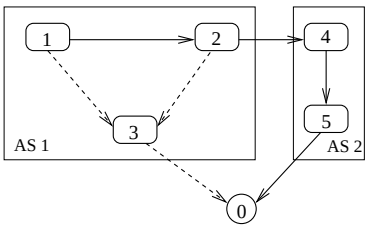

Fig. 3. A transient failure experienced by router 2 during the link between router 0 and router 3 is recovered. The dashed line represents the best path after the recovery event.

router cannot be transited to another iBGP router. For example, in Fig. 3, AS 1 has three iBGP routers. Before the link between routers 3 and 0 recovers, all routers within AS 1 select the path via AS 2 as their best path. After the recovery event, all routers within AS 1 use the recovery path, represented as dashed lines. Suppose that the recovery path sent by router 3 arrives at router 1 earlier than at router 2. After router 1 selects the recovery path as its best path, it cannot send the path to router 2 because of iBGP constraint. In this case, router 2 will lose its reachability to the destination until it receives the recovery path from router 3.

From the above examples, we can tell that the major reason contributing to transient routing failures is *single* route routing. In other words, only the best route can be exchanged between routers. However, BGP's loop prevention or routing policies can impact the path visibility, which leads to routing with partial route information. A straightforward solution is to enable a router to inform its neighbors multiple routes instead of the best one. However, multi-path routing scheme consumes too much resources, including the bandwidth and the storage spaces for the additional alternative paths. Another possible way to reduce the impact of transient failures on traffic is to use small MRAI timer or disable MRAI timer

because the duration of transient failures depends on MRAI timer. However, small MRAI timer may cause heavy updating message overhead on routers and lead to routing instability problem.

III. Routing with Backup Route Information

In this section, we present BRAP in detail. We start by providing an overview of our approach. Then, we describe the update messages, route computation and packet forwarding procedures.

A. Overview

Our objective is to design a survivable routing that can locally provide alternate routes upon the occurrence of failures. To this end, we examine the potential recovery paths that we can add to BGP. The design of BRAP is centered around the possibility of routing with complete route information.

In theorem, the maximum number of possible paths at each BGP router equals to the number of its neighbors. This motivates us to design a full route aware routing protocol, i.e., a routing protocol taking advantage of complete routing information from neighbors. In order to implement full route aware routing, a router should be enabled to advertise an alternate path if its best path is not allowed to advertise due to loop prevention or routing policies. Thus, the general idea for BRAP is as follows:

> *A router should have such capability to advertise following policy compliant paths in addition to the best path: 1) a reverse path to its primary neighbor; and 2) a loop-free alternate path, defined as a temporary backup path, to its upstream neighbors.*

As long as a policy compliant path exists, BRAP can find and use the path to deliver data upon a failure occurring. For example, in Fig. 1, router 1 has reverse path (1 0) and advertises the path to its primary neighbor 3. Thus, router 3 has the best path (3 0) and the backup path (3 1 0). When the link between router 3 and 0 fails, 3 still has the reverse path so that it does not experience any transient routing failure. In Fig. 2, when the link between router 0 and 3 recovers, router 1 will switch to the recovery path and advertise a withdrawal to 2 due to its routing policy. In BRAP, instead of advertising a withdrawal to 2, router 1 advertises an alternative path. As a result, router 2 still has an available path via router 1 to the destination.

Hence, we find that BRAP can address transient routing failures under both failover events and recovery events within an unified framework. Note that *different* routing information should not cause *inconsistent* route, which might give rise to other problems, such as routing loops. For example, in Fig. 1 if the link between router 0 and router 3 fails, router 3 can switch to the reverse path $(3\ 1\ 0)$. However, during this phase, router 1 still uses the path through router 3, which can lead to a routing loop between routers 1 and 3. Therefore, we need to design a more sophisticated routing protocol that supports routing with full path information without causing any routing loop.

B. Identifying Reverse Neighbors

Based on the best path, a BGP router can identify its primary neighbor and backup neighbor. The primary neighbor is the next hop of the best path. If a neighbor provides an alternate path, the neighbor is characterized as a backup neighbor. However, there is no mechanism for a BGP router to know its upstream neighbor. In BRAP, we use an Alternate Route Update message (ARU) containing a reverse path to identify a router's upstream neighbor and reverse neighbor. A router u can identify its reverse neighbor and upstream neighbor as follows. Suppose that u is the primary neighbor of v. If v has a reverse path, v sends u an ARU message containing the path. Upon receiving the ARU, u considers v as its reverse neighbor. If v does not have any reverse path, or propagation of the reverse path violates v's export policy, v advertises u an ARU message with empty path. Using this message, u can identify v as its upstream neighbor.

C. Route Selection

1) The Reverse Path: BRAP uses BGP's route selection function to select the reverse path among all possible paths except the best path. In particular, the rank of reverse paths should be the least preferable among all available paths. The intuition behind this is that using reverse paths might cause routing loops. Therefore, the reverse paths should be much less frequently used than other paths. Note that the

reverse path should be a loop-free path, i.e., it does not contain the primary neighbor's AS number or cluster ID. If there are multiple alternative paths available, the reverse path is not necessarily the disjoint path with the best path, which is different with R-BGP implementation [10]. As we described before, R-BGP only provides failure recovery on AS level. Further, AS paths are too coarse to identify real disjoint paths. We will show the reason that BRAP does not require disjoint path in Section III-F.

2) Temporary Backup Path: Suppose that upon receiving a update, a router switches to the new path containing in the update. However, the new path cannot be advertised to the router's upstream neighbor due to the router's routing policies or iBGP constraint. Instead of advertising a withdrawal to the upstream neighbor, the router generates a new type message, Temporary Route Update message (TRU). A temporary backup path, which is one of the router's available paths and is allowed to advertise to the upstream neighbor, is inserted into the message. In BRAP, the temporary backup path, like the reverse path, is the least preferable among all available paths.

D. Reaction to Topology Changes

Upon receiving a withdrawal message, In BRAP, a router u needs to recompute the best path, and advertises the path if the best path changes, just like a BGP router. Assume that u has at least one reverse path. Once u loses its all paths except the reverse paths, it switches to one of the reverse paths from v. To simplify our analysis, we define the neighbor from which the selected reverse path is learned as the *primary reverse neighbor*. In this case, v is the primary reverse neighbor of u. Router u needs to perform the following tasks:

1) Selecting a new best path among all reverse paths.
2) Advertising the current best path to all neighbors except the primary reverse neighbor v.
3) Generating a reverse path notification message (RPN), which contains current best path, and sending the message to the primary reverse neighbor v.
4) Selecting a new reverse path from a neighbor w.
5) Generating an ARU message, which contains the new reverse path if the path does exist, or an empty path if there is no such path.
6) Advertising the ARU message to previous primary reverse neighbor v.

Here, we design a new message RPN, which is used to notify the primary reverse neighbor to update its routing table once a failover event leads to use the reverse path. The purpose of the ARU message is to notify previous primary reverse neighbor v the new reverse path. As a result, previous primary reverse neighbor v will become primary neighbor, and u becomes a reverse neighbor of v.

Upon receiving the RPN message from u, v needs to perform:

1) Removing the current best path, which is learned from neighbor u.
2) Using the route containing in the RPN message as current best path.
3) Advertising the best path to all neighbors except u.
4) Selecting a new reverse path and advertising an ARU message containing the new reverse path to its current primary neighbor.

Next, we consider the case that router u receives an announcement which contains a new path. Suppose that before receiving the message router u is the primary neighbor of v. Thus, after receiving the new path, u has at least two paths: the new path and previous best path before the event. Suppose that u chooses the new path as its best path, but the path is not allowed to advertise to v due to its routing policies or iBGP constraint. Router u selects a temporary backup path among all paths except the current best path, which is allowed to send to v, and propagates a TRU message containing the path to v. After v receives the TRU, it will forward traffic through the temporary backup path before it obtains a new path. Once u receives a update from v related to the destination, including BGP announcement or ARU, u will send a withdrawal message to v to withdraw the temporary backup path.

E. Packet Forwarding

Using the reverse path or temporary backup path might give rise to a routing loop or violate routing policies or iBGP constraint. In BRAP, we use following techniques:

- Interface-specific forwarding technique, which is described in work [31]. Based on this technique, a router can detect when its primary neighbor uses a reverse path. In particular, the reverse neighbor detects if it is receiving traffic from a neighbor to whom it would forward that traffic. If traffic fails the reverse forwarding check, then the traffic is forwarded along the reverse path. Note that this method can avoid one hop forwarding loop.
- MPLS-based solution. Our previous work [27] has shown that only intradomain routers can experience packet loss during recovery events. Therefore, an intradomain router can use MPLS to forward packets to its primary neighbor along the temporary backup path. The MPLS protection LSP can be easily established by the primary neighbor and the upstream neighbor because they are in the same AS.

By using interface-specific forwarding during failover and recovery events, we use the first packet to synchronize routing tables between the primary neighbor and reverse neighbor. As a result, packets are not dropped due to lack of routing entry or forwarding loops. This method can achieve sub-millisecond failure recovery. The limitation of this method is that we need to modify the forwarding mechanism.

F. Properties of BRAP

By exchanging an ARU message, a router can identify the relationship between its neighbors. We have the following lemma:

Lemma 1 *if a router i has a reverse path to a destination, some upstream neighbors along the reverse path must have a path that is not via router i.*

Proof: By contradiction, we assume that a upstream neighbor j of router i has a path via i to a destination d. Suppose the path at router j is $(j \ldots, i, \ldots, d)$. When router j advertises this path to i, suppose that j implements Sender Side Loop Detection (SSLD), then router j is not able to advertise this path to router i. Therefore, router i does not have the reverse path, which is a contradiction. ■

This lemma implies that it is possible for a router to use its reverse path to reach the destination.

BRAP implements backup route aware routing by adding reverse paths and temporary backup paths. Having full path information, BRAP can achieve fast failure recovery.

Theorem 1 *For a given network with a given destination, if every primary router, which has one or more reverse neighbors, has at least one reverse path, the network can tolerate one single link failure occurring in the network.*

Proof: Every router has at least two available paths to a destination: the best path and the reverse path. We are going to prove that one single link failure cannot impact the both paths. Suppose that u has two paths: the best path $(uv_iv_{i-1} \ldots v_0)$ and the reverse path at$(uw_jw_{j-1} \ldots v_0)$, where v_0 is the destination. Suppose that the two paths share at some routers $v_k = w_{k'}$. Therefore, the two paths are via router v_k, and their subsequent paths from u to v_k are disjoint. So the network can tolerant any link failure between u and v_k. Let's continue to see the case that the link between v_k and v_{k-1} fails. Since v_k does not have any reverse path from its upstream neighbors v_{k+1} and $w_{k'+1}$, it must have a reverse path from another upstream neighbor. According to Lemma 1, the reverse path is not via router v_k so that router v_k can use the reverse path to recover the failure. ■

As we described before, BRAP does not require that the reverse paths must be disjoint with the best path. For example, in a network shown in Figure 4, all primary routers 3 and 5 have reverse paths. Even though router 3's reverse path (3 2 4 5 0) and best path (3 5 0) share link (5 0), the network can tolerate any link failure in the network. Note that router 5 only has one reverse path, which is from router 1. Router 3 and 4 cannot advertise their reverse paths to 5 because those paths are not loop-free paths. If router 1 considers path (1 2 3 5 0) as the reverse path, it cannot send the path to 5 because the path fails the loop check. As a result, router 5 does not have any reverse path and the network cannot tolerate link (5 0) failure. In this case, the primary neighbor (router 5) and the upstream neighbor (router 1) could negotiate the

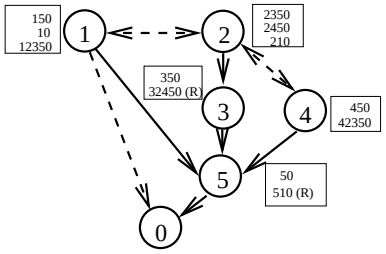

Fig. 4. Example used for illustrating how BRAP can tolerate any link failure in a network even though the reverse path and the best path might share some common links. In a routing table, an entry with "R" represents a reverse path received from a reverse neighbor.

reverse path selection. Based on the theorem, we can protect a network, which could be a subset of the whole network, for example, an AS in the Internet. To tolerate any link failure occurring within an AS, we need to make sure that all routers inside the AS satisfy the theorem. In this case, the overhead for having reverse paths within an AS is much small.

G. Protocol Overhead

Next, we examine the number of messages generated by our approach. Here, we only study the number of messages exchanged during a failover event. During a recovery event, BRAP replaces a withdrawal message, which is generated due to loop avoidance or routing policies, with an announcement. Therefore, our approach sends the same number of messages as BGP does during a recovery event.

Suppose that a primary neighbor A has n neighbors. Suppose that primary neighbor A uses the reverse path from neighbor B as the best path upon a failover event occurring. The reverse neighbor B has m neighbors, where $n, m > 2$. The number of messages N sent by both routers is

$$N = n + m + 1$$

We first look at the number of messages sent by primary neighbor A. When A uses the reverse path from B to reroute traffic upon a failover event, it sends at most $n + 1$ messages, which include one RPN message, one ARU message, and $(n - 1)$ announcements. Second, we look at the number of messages sent by reverse neighbor B. Neighbor B sends at most m messages, including $m - 1$ announcements about the new best path and one ARU message. Third, we look at the number of messages sent by A when A receives a RPN from B. In this case, A does not advertise any route. Therefore, A and B will send $n + m + 1$ messages.

Let us look at the number of messages generated by BGP in the same event. BGP will send n withdrawal messages, m announcement about the new best path generated at B, and $n - 1$ announcement about the new best path advertised at A. Overall, BGP will generate $2n+m-1$ messages. We find that the number of messages generated by our approach is much less than BGP does. Our simulation in the next section also verifies this observation.

IV. Practicality of Deploying BRAP in a Typical BGP System

In this section, we discuss the practicality of deploying BRAP in a typical BGP system. A typical BGP system means that every router in the system applies *typical routing policies* and the hierarchical iBGP configurations are deployed. In particular, the export routing policies are typically guided by the *no-valley* routing policy, in which an AS does not export its provider or peer routes to its providers or peers. The import routing policies are guided by the *prefer-customer* routing policy, in which each AS prefers its customer routes over its peer or provider routes. In a hierarchical iBGP structure, there are route reflectors, and a set of *edge routers*, which are router reflectors' clients.

A. Deploying BRAP to Interdomain Routers

We first consider deploying BRAP to interdomain routers. In our previous paper [27], we show that only non-tier 1's interdomain routers can experience packet loss during failover events, and no interdomain routers experiences packet loss during recovery events. Therefore, we study which ASes could be the reverse neighbor for a given AS with respect to a given destination.

Theorem 2 *In a typical hierarchical eBGP system, the reverse neighbor of an interdomain router v must be a upstream router inside a provider.*

Proof: We prove this by contradiction. We assume that AS v is a customer or peer of AS u. Suppose that AS u has a reverse path $P =$

$(vv_kv_{k-1}\dots v_1)$ to reach the destination. According to Lemma 1, the path is a loop-free path. Furthermore, the path must not violate no-valley policy. Since the AS relationship between AS u and AS v is provider-to-customer or peer-to-peer, path (v_iv_{i-1}) at AS v must be a customer link for any $i = k, k-1, \dots, 2$. In this case, AS v has at least two paths: the best path from AS u and the customer path from v_i. According to prefer-customer routing policy, AS v must select v_i as its primary neighbor, which is a contradiction. ■

Thus, an AS can deploy BRAP in its interdomain routers peering with its providers. Since the reverse paths only come from providers' routers, customers are willing to require their providers to provide fast failure recovery service – the reverse paths. At the same time, for the providers, such service is limited to subscribed customers so that they are also willing to advertise their reverse paths.

Since tier-1 ASes do not have providers, we have following property for tier-1 ASes:

Corollary 1 *In a typical hierarchical eBGP system, tier-1 ASes do not have reverse neighbors from their interdoamin routers.*

We will show in the next section that tier-1 ASes's reverse neighbors locate at their own network so that their reverse paths come from their routers.

Non-tier ASes could be transit ASes and stub ASes. According to Theorem 2, transit ASes will obtain the reverse paths from their providers. For stub ASes, however, they do not need to deploy BRAP. For the prefixes from other ASes, stub ASes and their providers do not have primary-to-reverse relationship because stub ASes cannot provide any reverse path to their providers due to no-valley policy. Therefore, in the Internet, only the transit ASes need to deploy BRAP. Our measurement shows that the number of transit ASes is much less than the number of stub ASes. Only a small number of ASes taking advantage of BRAP can achieve survivable routing for the Internet.

B. Deploying BRAP to Intradomain Routers

As we described before, Intradomain routers can experience packet loss during both failover and recovery events. Thus, an AS needs to enable BRAP at its intradomain routers. An AS can take advantage of MPLS or IP tunnels to forward packets when a reverse path or temporary backup path is used, which can be easily implemented inside a network.

Large ASes typically have a fully meshed or a hierarchical iBGP structure. Here we focus on a hierarchical iBGP structure. In a hierarchical iBGP structure, there are route reflectors, and a set of *edge routers*, which are router reflectors' clients. An edge router could be an *access router* that connects to a customer network, or a *peer router* that connects to a peer network.

We first discuss deploying BRAP to prevent packet losses from failover events. For tier-1 ASes, their reverse neighbors are their own routers. That means, to prevent the occurrence of transient failures due to failover events, tier-1 ASes only need to enable FARP at their own routers. In this case, tier-1 ASes can take advantage of MPLS or IP tunnels to forward packets when a reverse path or a temporary backup path is used.

For non-tier 1 transit ASes, they can implement BRAP among their routers to avoid transient failure like tier-1 ASes. On the other hand, they need to consider to deploy BRAP across eBGP routers, which connect to their providers. In this case, the routers at their providers will provide reverse paths. Therefore, the cooperation is required between the transit ASes and their providers.

Our previous work [27] has shown that if every AS follows the typical routing policies, no AS would experience transient routing failures among its eBGP routers during the recovery events. In this case, only iBGP routers within an AS might experience transient failures. This implies that the overhead of using BRAP to avoid transient failures during recovery events is much lower than failover events. In BRAP, two routers need to agree with their route preference for the temporary backup path. Therefore, network operators can easily implement the required route preference for the temporary backup paths.

C. Which Destination Prefix Should be Protected?

We have discussed the guidelines for deploying BRAP at a typical BGP system. But which destination prefix should be protected by BRAP? Let us go back to the example shown in Fig. 1 at Section II. Suppose that each node in the figure represents an

AS. AS 1 and AS 2 are AS 3's providers, while AS 1 and AS 2 are peer-to-peer. AS 1 and As 3 are AS 0's direct providers, and AS 2 is AS 0's indirect provider. Furthermore, the destination is advertised to two ASes. From above discussion, we know that only transit ASes should deploy BRAP to protect the destination. Motivated by this example, we find that an AS can enable BRAP to protect a prefix which is a *multi-homed* prefix and *originated by the AS's customer or indirect customer*.

V. Evaluation

In this section, we evaluate the effectiveness of BRAP. We first use routing data from the Internet to examine full path routing provided by BRAP. Second, we evaluate performance of BRAP by using a simulation.

A. Route Coverage

We first examine routing with partial route information in BGP to understand the improved resilience by using BRAP. We measure *route coverage*, which is defined as the ratio between the number of an AS's neighbors and the number of available paths seen in BGP. Note that the metric is measured on AS level. We use two datasets, which are collected at the same day (July 17, 2006), to infer the AS topology. One dataset is BGP tables from Oregon RouteViews, and another dataset is AS path information from 60 ASes' Looking Glass servers. The second dataset is to use to provide a more complete view of the AS-level topology. Since BRAP does not take effect for single-homed ASes, we discard all single-homed stub ASes. Of 23284 ASes in our dataset, there are 7589 single-homed stub ASes. As a result, the final AS topology consists of 15695 ASes, including 3171 transit ASes and 12524 multi-homed stub ASes.

Next, we describe the method to infer available paths. We first infer AS relationships, and then apply conventional policies for selecting and exporting routes to construct all available paths. There are several algorithms which can be used to infer AS relationships [8] [26] [3]. Here, we choose the one described in [8]. We assume that each AS originates a *single* destination prefix and each AS selects and exports routes based on AS relationships. That is, each AS always prefers the routes from customers over routes from providers or peers, and obeys no-valley policy to export routes. Given an AS as a destination, we derive available paths between each AS and the destination AS. After that, for each AS, we select its best route according to prefer-customer policy. If there are multiple paths coming from customers, we select the one with shortest AS path length. After finishing inferring the best path, we continue to infer the reverse neighbor and reverse paths.

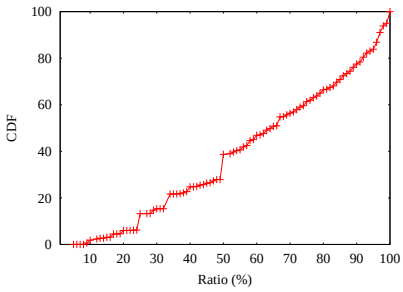

Fig. 5. Cumulative distribution function for the ratio between number of neighbors and available paths in BGP.

In Fig. 5, we show the cumulative distribution function for route coverage. The x axis represents the ratio. We observe that in BGP more than 40% of ASes only know less than half of available paths from their neighbor so that they do not have full path information. Therefore, BRAP significantly increases route coverage and provides full route information to recover transient failures.

Next, we measure performance of BRAP in term of reverse nodes in a reverse path, which reflects the convergence delay. In Fig. 6, we observe that more than half of reverse paths only involves 1 reverse neighbor. That means, the next hop neighbor (reverse neighbor) has at least one alternate path so that the failure duration is quite short. We also observe that the chance to involve more than 5 reverse neighbors is quite small.

B. Simulation Result

In this section, we use simulation to measure the performance of BRAP in terms of the number of messages, transient failure duration and convergence time. We implement BRAP based on the event-driven BGP simulator simBGP [1]. In our simulation, each node has a random processing

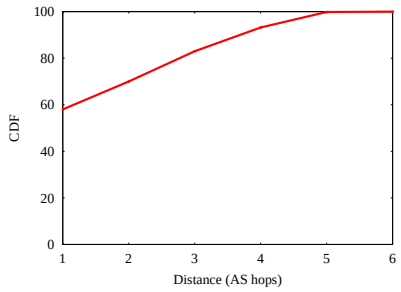

Fig. 6. Cumulative distribution function of the number of reverse neighbors in a reverse path.

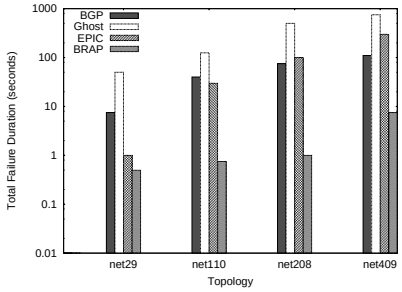

Fig. 7. The duration of transient forwarding failures.

delay with uniform distribution between [0.001, 0.01] millisecond. Each link is assigned bandwidth 100MB and a queuing delay uniformly distributed between [0.01, 0.1] millisecond. The MRAI timer for eBGP sessions is set to 30 seconds and that for iBGP sessions is 5 seconds. Note that the MRAI timer is peer based. In our simulator, when a node is ready to send routing message to one of its peer and if the MRAI timer for this peer is not set, we will artificially block the message with a duration uniformly distributed between [0,MRAI]. That is, the MRAI timer is assumed to be set by the background routing messages. We also assign 200 msec to each node as the FIB updating delay.

The simulations are performed based on the AS level topologies provided by work [23] and router level topologies. That is, an AS consists of both iBGP and eBGP routers. In each simulation, we pick one AS to originate a destination prefix. When every router reaches stable states, we break one of the egress link of the origin AS. Because the ASes in the topologies are densely connected, the AS is still connect to the network, which triggers a failover event. We repeat the operation for every

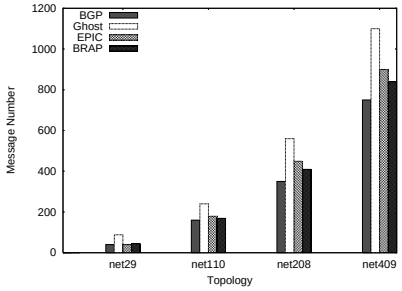

Fig. 8. Number of updates generated by BGP, Ghost, EPIC and BRAP during Failover events

egress link of an AS for 5 times and perform the same process for every AS in a network. Besides, we also investigate the scenarios where the ghost-flushing and EPIC schemes are employed.

We first compare the duration of transient failure in BRAP, BGP, Ghost and EPIC. We then examine the number of updates generated by them. As shown in Fig. 7, the X axis represents the topology where the simulations are performed. For example, "net 29" means the AS topology contains 29 AS nodes and so on. From Fig. 7, we observer that BRAP indeed provides fast transient failure recovery. Fig. 8 shows the message numbers generated by BRAP, BGP, Ghost and EPIC. As expected, BRAP generates less routing messages than BGP, Ghost and EPIC. From our simulation results, we find that BRAP can significantly accelerate the convergence speed and eliminate transient failures during the convergence process.

VI. Related Work

Significant works have been done to achieve fast rerouting for IGP. Recent work shows subsecond IGP convergence time can be achieved by fine tuning parameters of IGP [25]. MPLS based approaches use pre-computed backup paths to reroute around failures immediately after detecting link failures. However, this method is usually done in a centralized manner. Nelakuditi *et. al* use interface specific forwarding tables to achieve fast rerouting [19], which requires a new algorithm for forwarding table calculation. Narvaez *et. al.* focus on link state protocols and propose a local restoration algorithm. On the contrary, our work focus on fast rerouting during BGP convergence.

Previous studies have considered approaches to accelerate BGP route convergence, including ghost-flushing [6], BGP-RCN [21] and EPIC [30]. However, those approaches can not achieve non-stop forwarding, and even exacerbate transient routing failures.

Bonaventure *et. al* propose that a BGP router selectively establish protection tunnels to the relevant routers which can feed alternative routes to these routers in case of routing failures [4]. However, the ability that this approach provides is limited because a router needs to establish a tunnel with a right peer that can provide the alternative path.

Kushman et al [10] propose a mechanism that protects data forwarding from failover events by providing failover paths. The limitation of the proposal is that it only focuses on providing fast recovery upon failover events. None of the protocols we overview above are able to prevent the presence of a wide range of transient routing failures, but are rather focused on a particular failure.

VII. Conclusion

This paper presents a solution to provide a backup route aware routing protocol to fast failure recovery. The heart of our approach is the capability of redirecting traffic to a preestablished path that cannot be disrupted by routing dynamics. By using reverse paths and temporary backup paths, our approach can maximize the number of available paths so that it can provide fast failure recovery and more reliable end-to-end path performance.

References

[1] Simple BGP Simulator. http://www.bgpbista.com/simbgp.shtml/.

[2] Agarwal, S., Chuah, C., Bhattacharyya, S., and Diot, C. The Impact of BGP Dynamics on Intra-Domain Traffic. In *Proceedings of ACM SIGMETRICS* (June 2004).

[3] Battista, G. D., Patrignani, M., and Pizzonia, M. Computing the Types of the Relationships Between Autonomous Systems. Technical Report RT-DIA-73-2002, Dipartimento di Informatica e Automazione, University Roma Tre, 2002.

[4] Bonaventure, O., Filsfils, C., and Francois, P. Achieving Sub-50 Milliseconds Recovery Upon BGP Peering Link Failures. In *Proceedings of ACM CoNEXT* (2005).

[5] Boutremans, C., Iannaccone, G., Bhattacharyya, S., Chuah, C., and Diot, C. Analysis of Link Failures in an IP Backbone. In *Proceedings of ACM SIGCOMM Internet Measurement Workshop* (November 2002).

[6] Bremler-Barr, A., Afek, Y., and Schwarz, S. Improved BGP Convergence via Ghost Flushing. In *Proceedings of IEEE INFOCOM* (2003).

[7] Feamster, N., Andersen, D., Balakrishnan, H., and Kaashoek, M. Measuring the Effects of Internet Path Faults on Reactive Routing. In *Proceedings of ACM SIGMETRICS* (San Diego, CA, June 2003).

[8] Gao, L. On Inferring Autonomous System Relationships in the Internet. *IEEE/ACM Transactions On Networking 9*, 6 (December 2001).

[9] Hengartner, U., Moon, S., Mortier, R., and Diot, C. Detection and Analysis of Routing Loops in Packet Traces. In *IMW* (2002).

[10] Kushman, N., Kandula, S., Katabi, D., and Maggs, B. R-BGP: Staying Connected In a Connected World. In *4th USENIX Symposium on Networked Systems Design & Implementation* (2007).

[11] Kvalbein, A., Hansen, A. F., Cicic, T., Gjessing, S., and Lysne, O. Fast IP Network Recovery using Multiple Routing Configurations. In *INFOCOM* (2006).

[12] Labovitz, C., and Ahuja, A. The Impact of Internet Policy and Topology on Delayed Routing Convergence. In *Proceedings of IEEE INFOCOM* (Anchorage, Alasks, April 2001).

[13] Labovitz, C., Ahuja, A., Bose, A., and Jahanian, F. Delayed Internet routing convergence. *IEEE/ACM Transactions on Networking 9*, 3 (June 2001), 293–306.

[14] Labovitz, C., Ahuja, A., and Jahanian, F. Experimental Study of Internet Stability and Backbone Failures. In *Proceedings of FTCS* (1999).

[15] Labovitz, C., Malan, G. R., and Jahanian, F. Internet Routing Instability. *IEEE/ACM Transactions on Networking 6*, 5 (1998), 515–528.

[16] Lee, S., Yu, Y., Nelakuditi, S., Zhang, Z.-L., and Chuah, C.-N. Proactive vs Reactive Approaches to Failure Resilient Routing. In *Proceedings of IEEE INFOCOM* (2004).

[17] Markopoulou, A., Iannaccone, G., Bhattacharyya, S., Chuah, C., and Diot, C. Characterization of Failures in an IP Backbone. In *Proceedings of IEEE INFOCOM* (2004).

[18] Narvaez, P., Siu, K.-Y., and Tzeng, H.-Y. Local Restoration Algorithm for Link-State Routing Protocols. In *ICCCN* (1999).

[19] Nelakuditi, S., S. Lee, Y., and Yu, Z.L, Z. Failure Insensitive Rerouting for Ensuring Service Availability. In *IWQoS* (2003).

[20] Oregon. http://www.antc.uoregon.edu/route-views/.

[21] Pei, D., Azuma, M., Massey, D., and Zhang, L. BGP-RCN: Improving BGP Convergence Through Root Cause Notification. *Computer Networks and ISDN Systems 48*, 2 (2005), 175–194.

[22] Pei, D., Zhao, X., Massey, D., and Zhang, L. A Study of BGP Path Vector Route Looping Behavior. In *ICDCS* (2004).

[23] Premore, B. Multi-AS Topologies from BGP Routing Tables. http://www.ssfnet.org/Exchange/gallery/asgraph/index.html.

[24] Roughan, M., Griffin, T., Mao, Z. M., Greenberg, A., and Freeman, B. Combining Routing and Traffic Data for Detection of IP Forwarding Anomalies. In *Proceedings of ACM SIGCOMM NeTs Workshop* (2004).

[25] Shaikh, A., and Greenberg, A. OSPF Monitoring: Architecture, Design and Deployment Experience. In *Proceedings of USENIX Symposium on Networked Systems Design and Implementation* (March 2004).

[26] Subramanian, L., Agarwal, S., Rexford, J., and Katz, R. H. Characterizing the Internet Hierarchy from Multiple Vantage Points. In *Proceedings of IEEE INFOCOM* (2002).

[27] WANG, F. Internet Routing and Its Transient Behavior. *Ph.D Thesis University of Massachusettes*, Amherst (2006).

[28] WANG, F., MAO, Z. M., WANG, J., GAO, L., AND BUSH, R. A Measurement Study on the Impact of Routing Events on End-to-End Internet Path Performance. In *Proceedings of ACM SIGCOMM* (2006).

[29] XU, W., AND REXFORD, J. MIRO: multi-path interdomain routing. In *SIGCOMM* (2006).

[30] ZHENHAI, J. C. Limiting Path Exploration in BGP. In *Proceedings of IEEE INFOCOM* (2005).

[31] ZHONG, Z., KERALAPURA, R., NELAKUDITI, S., YU, Y., WANG, J., CHUAH, C.-N., AND LEE, S. Avoiding Transient Loops Through Interface-Specific Forwarding. In *IWQoS* (2005).