Resilient Routing under Hierarchical Automatic Addressing

Yang Song*, Lixin Gao*, and Fujikawa Kenji**

*Department of Electrical and Computer Engineering, University of Massachusetts, Amherst,
Email: ysong, lgao@ecs.umass.edu
**New Generation Network Research Center, National Institute of Information and Communications Technology
Email: hudikaha@nict.go.jp

***Abstract*—BGP table size and update rate have increased dramatically in the last decade. The rate of the growth poses great demand on memory size and CPU speed of the router control processor. Hierarchical Automatic Addressing (HAA) reduces BGP table size and update rate by enabling prefix aggregation for multihomed ASes. However, the aggregation blocks routing information of individual ASes, and causes the routing system fails to react to failures. In this paper, we propose a resilient routing protocol referred to as *Routing with Detour (RD)* to address the problem. RD provides resilient routing under HAA, and it reacts to failures in a timely manner without modifying BGP. Our experiments with realistic AS topology show that RD has slight impact on HAA's performance. More precisely, HAA with RD can reduce BGP table size and update rate by more than 90%.**

I. Introduction

Border Gateway Protocol (BGP) is used to exchange reachability information among routers in the Internet. BGP table size has grown rapidly in the last decade [2] [9] [3], and approaches 330,000 entries in 2010. In addition, BGP update rate increases at a much faster pace than BGP table size [8], and the peak update rate is up to 1000 times higher than the daily average [10]. However, routers' memory size and CPU speed might not be able to meet the demand of the fast growth. The global routing system, especially the core routers, faces significant scalability problems that the growth brings.

IP prefix aggregation can address some of the scalability issues. Contiguous IP addresses or prefixes can be aggregated into one prefix, so that fewer prefixes will be announced to the Internet, and thus less entries need to be installed in BGP tables. The aggregation also reduces update messages propagating to the Internet, which have been shown as one of the major contributors for the scalability problems [12].

However, address aggregation can not be performed when ASes are multihomed. Since prefixes must be announced through all the providers, it can be aggregated into at most one provider's prefix, from which the customer inherits the prefix. Usually, even the provider that is able to aggregate the prefix would not conduct the aggregation, because the longest prefix match will prevent the aggregated prefix from being chosen. As transit fee gets cheaper, more and more networks subscribe to multiple providers, and multihoming becomes one of the major causes of BGP table growth [2].

Hierarchical Addressing Allocation (HAA) [4] has been proposed to aggregate prefixes under multihoming. Basically, it allows one host to possess multiple IP addresses. Thus, if an AS is multihomed, it can inherit one prefix from each of its providers. As a result, all the prefixes are able to be aggregated, and BGP table size can be reduced significantly.

However, HAA introduces routing problems. BGP is designed to maintain routing information of each destination prefix. With HAA implemented, BGP only keeps the information of aggregated prefixes. Thus, it fails to react to the network events that happen to individual destination prefixes.

In this paper, to address these challenges under HAA, we propose a routing protocol called Routing with Detour (RD). Without changing BGP, we run a light-weighted routing process to work together with BGP. It can reroute traffic when BGP fails to react to failures because of aggregation.

We prove that, by implementing RD with HAA, Internet routing is robust under one or multiple network events. Further, our experiments with the realistic AS topology show that, RD only has slight impact on BGP table size and update rate. Only 12.4% ASes need to install routing entries for RD, and only 127 ASes need to install more than 100 RD routing entries. Less than 1% of the links may generate an update message to maintain RD. Comparing to the current Internet setting, HAA with RD can reduce BGP table size and update rate by 99.1% and 90.0% for most of the ASes.

More interestingly, we show that, implementing HAA with RD on stub network only can significantly improve the scalability of the Internet. This method is easier to implement comparing to the global deployment, and only consumes 33 extra /8 prefixes. With this setting, BGP table size and the update rate can achieve 93.3% and 34.3% reduction respectively.

The outline of the paper is as follows: In Section II, we introduce HAA and explain why there are problems in the Internet routing under HAA. Section III presents the details of our routing method. In Section IV, we show the experimental results. Section V is the conclusion.

II. HAA and Internet Routing

Here, we focus on the interdomain routing under HAA. How to conduct routing within an AS and how to automatically

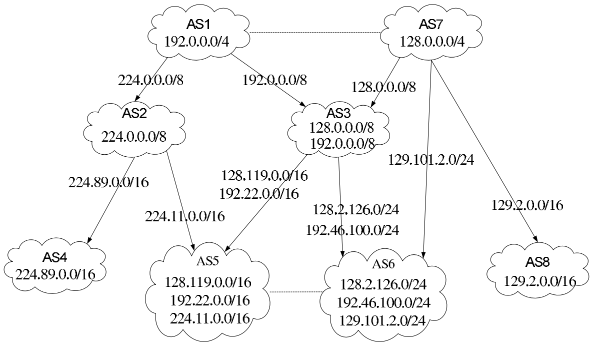

Fig. 1. Global Deployment

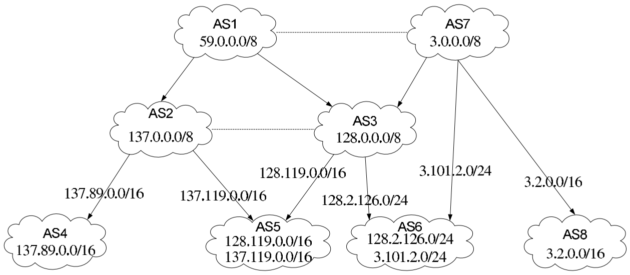

Fig. 2. Stub Network Deployment

assign multiple IP addresses to a host are introduced in [4] and [5].

A. Global deployment and stub network deployment

In this paper, we show two ways to implement HAA: global deployment and stub network deployment. In global deployment, all the ASes inherit one sub-prefix from each provider, while in stub network deployment, only the stub networks are required to do so.

Figure 1 depicts the global deployment. Arrows indicate provider-customer relationship with arrowheads pointing to the customers, and dash lines indicate peer-peer relationship. Tier-1 ASes own prefixes that cover a large address space, and they assign sub-prefixes to their customers. Non-Tier-1 ASes inherit one prefix from each provider, and assign their customers one sub-prefix from each of these prefixes they inherit. Figure 2 depicts the stub network deployment. In this setting, only stub networks are required to get one sub-prefix from each of its providers.

B. Routing under HAA

Under HAA, multiple prefixes can be aggregated into one larger prefix. Thus, BGP updates are only able to operate on aggregated prefixes instead of on the more specific prefixes before aggregating. Thus, routing information for these specific prefixes is hidden. This cannot effect new link discovery and peer-peer link failure as proved in [11], but the failure between the provider and customer link cannot be recognized . The traffic to the customer AS would still be forwarded to the provider AS, and it would be dropped eventually. Even though there may exist available paths to the customer AS through other providers, no update message is announced to trigger these available paths. We call the failure that happens between a provider and a customer the *access failures*.

[7] proposes a method to solve this problem by announcing the failed prefix through other working providers. The action will add the failed prefix to all the routers in the Internet. It not only involves more BGP update messages, but also needs a long time to propagate the failure. We propose a new method to handle these problems.

III. Routing with Detour (RD)

A. Basic idea

Suppose there is an access failure between AS2 and AS5 in Figure 1. In this case, AS5 is still reachable through AS3. If we can build a backup path between AS2 and AS3, AS2 can still announce the same aggregated prefix, and reroute AS5's packets through AS3.

Inspired by this example, we propose a resilient routing under HAA using backup paths for multihoming ASes. Because the backup paths can be installed and maintained locally, they involve small overhead. To isolate from BGP, we run another routing process for backup paths. That is, in routing tables, there may exist two paths to reach the same destination, one using BGP and the other one using backup routing process. One bit of information in the packet header is used to indicate which path the packet should follow. If there is no access failure, packets use BGP. In case of access failure, the provider AS flaps the information bit, and let the traffic follow the backup path. Because the backup path entries are installed in routers' forwarding table, the method can react fast to failures and minimize packet loss. In our experiment, we show that installing and maintaining backup paths only have slight impact on BGP table size and update rate. Implementing HAA with RD, both of them can be reduced by more than 90%.

Next, we introduce how to apply RD in detail.

B. Routing with Detour (RD)

A multihomed AS, represented by d, should build a backup path for each of its providers. We call the provider that we build a backup path for the *Original Provider*, represented by AS_o. A backup path should be built between the original provider and another randomly chosen provider. We call the provider used for backup the *Backup Provider*, represented by AS_b. Backup routing process maintains backup paths between AS_o and AS_b.

The challenge of implementing RD is how to build backup paths and how to maintain backup paths under different network events. In this section, we introduce these operations.

1) Backup path establishment: In the following content, we introduce how to build one backup path for a pair of original provider and backup provider.

We build a backup path which is the same as the BGP best path between AS_o and AS_b. Thus we need to install backup path entries on the ASes that are on the BGP best path. We name AS d's prefixes inherited from AS_o and AS_b the Original Prefix (OP) and the Backup Prefix (BP) respectively. Then, the

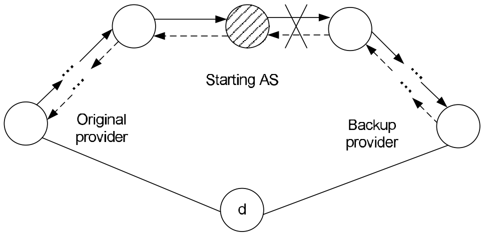

Fig. 3. Reaction to network events

prefix of the backup path entry should be equal to OP so that the packet can follow the backup path to reach d in case of failure. The operation details are as follows:

To initialize the establishment of a backup path, d should send a backup path announcement message to its original provider. The announcement message contains OP and BP.

After the original provider receives a backup path announcement message, it should start to establish the backup path. First, it should look up BP in its own BGP routing table, and add a backup path entry for OP with the same attributions. Then, the backup path announcement message should be propagated to next AS that is on the BGP best path to BP. The same operations should be repeated until d is reached. Note that, if multiple announcements for the same OP is received, only the first one needs to be propagated to the next hop AS.

In order to make backup path routing robust, the AS that builds a backup path entry needs two binding operations: first, the backup path entry for OP should be attached to the BGP entry for BP. Second, the backup path entry need to record the AS from which the backup path announcement message is sent. If multiple announcements for the same OP are received from different ASes, all the ASes' numbers should be recorded in the backup path entry.

C. Reactions to link failures

RD should be robust under any network event. In this section, we discuss this issue.

In Figure 3, we show a backup path between an original provider and a backup provider. Note that, it is the same as the BGP best path between the two providers. We use solid arrows to show the direction of backup path for OP, and use dash arrows to show the propagation direction of BGP update message for BP.

When a network event happens, starting from the *starting AS* which first changes the BGP best path to BP, a BGP update message will be propagated along the direction of dash arrows, and the ASes that receive the message would change their BGP entries. These changes can be used to indicate the failure of the backup path, and the backup path entries attaches to these BGP entries should be removed. That is, when a change happens in a BGP entry, all attached backup path entries should be removed. If the BGP best path change is announced to all the ASes which are attached to a backup path entry, no further operation is needed. Otherwise, a backup path announcement message should be initialized, and a new backup path should be built following the instructions in Section III-B1. If there is no new available path, a withdrawal message should be sent to d, and d should choose another backup provider.

So far, we only remove the unavailable backup path entries between the original provider and the starting AS. Next, we need to handle the backup entries between the starting AS and the backup provider. We call this part of the backup path the *upstream backup path*. On the upstream backup path, failures cannot be indicated by attached BGP entries. Thus, we use backup path withdrawal messages to remove them. We use the connection failure to the starting AS to indicate that the backup path is not available any more, and send a backup path withdrawal message following solid arrow to eliminate related backup path entries. After receiving the withdrawal message, the binding of the backup path and the failed AS is removed. If all the attached ASes are removed, the backup path entry should be removed, and a withdrawal message should be propagated to the next hop AS. Note that, there may exist multiple failures on the backup path, and each failure should be handled in the same way.

In summary, there are two kinds of network events that indicate a backup path failure: the change in BGP entries which are attached to backup path entries and the failures of the ASes which are attached to backup path entries.

With the operations above, a unique backup path can be established between a pair of original provider and backup provider, and the path is robust under one or multiple network events. The statements can be expressed with the following theorem. The proof can be found in [11].

Lemma 1: RD can build a backup path that is the same as the BGP best path between an original provider and a backup provider, and the backup path cannot be removed by intermediate withdrawal messages.

Lemma 2: When failures happen on a backup path, RD guarantees to remove all the entries of the backup path, even under multiple network events.

Theorem 1: RD builds only one backup path between an original provider and a backup provider, and the backup path is always the same as the BGP best path. It is true under any single or multiple network events.

IV. Evaluation

We already demonstrate the resilience of RD. However, the operation of RD increases routing table size and update rate. In this section, we show that the increase is slight. We also discuss QoS impact and IP addressing space usage, and our evaluation shows that it is feasible to implement HAA with RD.

A. Experiment setting

In order to perform realistic evaluation, we construct the Internet AS topology using the data collected by the RouteViews project [1]. We suppose that every AS would not change the commercial relationship with its neighbors, but IP prefixes can be rearranged. In global deployment experiment, we rearrange

IP addresses so that every customer inherits a sub-prefix from each of its providers. In stub network deployment, only stub networks are required to do so. In order to build backup paths, we need to know BGP best paths between every original provider and backup provider pair. In practice, this depends on complex routing policies which are usually private. However, it is commonly believed that routing decisions mainly depend on commercial relationship and break ties by selecting shorter routes over longer ones. Our experiments are conducted under these assumptions. The relationships between different ASes are identified using the method proposed in [6].

B. Impact on routing table size

1) Routing table size with global deployment: We first compare the current routing table size with the table size of HAA with RD. The overall comparison is shown in Figure 4. We can see that, the table sizes are significantly reduced. Currently, there are more than 330,000 prefixes in the routing tables. With global deployment, the routing table size is reduced by more than 81.3%. 90.2% of the ASes have a routing table with less than 3000 entries.

To analyze the impact of RD, we show the number of BGP entries and backup path entries separately in Figure 5, and both x-axis and y-axis are in log scale. ASes are sorted by the number of BGP entries (represented by the line with squares), and the number of backup path entries are drawn accumulatively on BGP entries (represented by pulses). Only 12.4% of the ASes need to install backup path entries. The number of backup entries ranges from 1 to 15,261. 18 ASes need to install more than 1,500 backup entries, and 127 ASes need to install more than 100 backup entries. We can see that the backup entries have small impact on the overall routing table sizes.

Most of the ASes that have a large number of backup entries are large ISPs (identified by pulses in Figure 5). We can see that, some large ISPs have a small number of BGP entries because of the address aggregation. This significantly release the work load of routers.

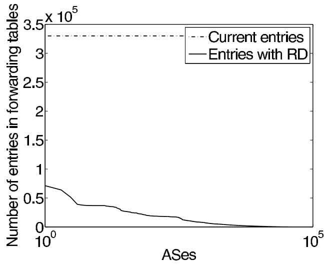

Fig. 4. Compare current routing table size and global deployment routing table size

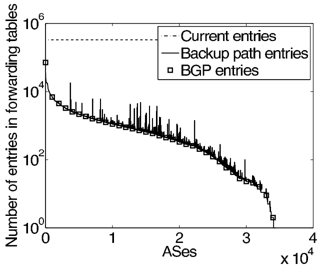

Fig. 5. Routing table size with global deployment

2) Routing table size with stub network deployment : In this case, only the stub networks' prefixes are eliminated from global routing tables, and thus, large ISPs still have a large number of BGP entries, and the distribution of backup entries looks different from global deployment. We compare the current table size with stub network deployment in Figure 6. Figure 7 shows the impact of RD with both x-axis and y-axis in log scale. In the two figures, ASes are sorted by the number of BGP entries, and backup path entries are drawn accumulatively on BGP entries. Comparing to the current table, the total table size is reduced by 93.3-96.3%. Among all the ASes in the Internet, only 10.0% of the ASes need to install backup path entries, and the number of backup path entries ranges from 1 to 7,394. Only 13 ASes have more than 1,300 backup path entries in their routing tables.

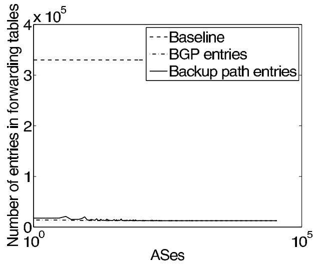

Fig. 6. Compare current routing table size and stub network deployment routing table size

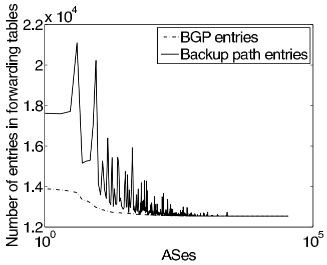

Fig. 7. Number of entries in routing table with stub network deployment

C. Impact on update rate

There are two kinds of network events that may generate update messages, link update and prefix update. Next, we discuss them separately.

1) Link update message: In the current Internet, network events that happen on each link might propagate to all the ASes. Suppose network events are uniformly distributed on links, then the link update message reduction will be equal to the percentage of links whose update messages have been blocked.

With global deployment, only peer-peer link events will be announced to the Internet. Depending on the location of ASes, the number of peer-peer links it receives updates from is different. Our experiments show that, 97% of the ASes will only receive BGP update message from less than 10% of the links.

On the other hand, maintaining backup paths only involves a small volume of update messages. In the worst case scenario, any network event happening on a backup path can be propagated to each AS on the backup path. The results show that only 12.4% of the ASes might receive backup path update messages, and the percentage of links which may send update messages to these ASes is less than 1%.

In stub network deployment, the network events that happen on the links between the stub networks and their providers cannot be propagated. That is, 34.3% of the links would not propagate update message. Only 10% of the ASes might receive backup path update messages, and the percentage of links which might send update messages to these ASes is less than 0.7%.

2) Prefix update message reduction: The failure or recovery of a prefix may propagate to all the ASes. There are 330,000 prefixes in current BGP tables, which indicates that each AS can potentially receive update message from 330,000 prefixes. After implementing HAA with RD, the number of prefixes that an AS may receive update message from is reduced. Suppose the probability of network events are uniformly distributed on all the prefixes. The number of AS update messages received by each AS is reduced by more than 90% in both deployments. The events happening to prefixes may also trigger update messages to maintain backup paths. According to our experiments, in both deployments, less than 12.4% of the ASes would receive update messages of backup paths, and most of the ASes would only receive update messages from less than 10% of the ASes.

D. Impact on AS path length and IP addressing space

If packets are forwarded through backup paths, AS path length would increase. It is important to understand how many extra ASes the re-routing packets would pass. In both global and stub network deployment, more than 93.7% of the backup paths only involve three or less extra ASes. Thus, most of the backup paths are short, and have limited effects on the rerouting packets.

We also compute how many IP addresses HAA with RD would consume. Multiplying the current IP addressing space an AS consumes by the number of prefixes each AS inherits from its providers, we obtain the address space usage under HAA with RD. The result shows that, stub network deployment only needs extra 5.5×10^8 IP addresses in total. The addresses can be provided by 33 /8 prefixes. However, in global deployment, 1.3×10^{10} IP addresses will be needed. The total number of addresses IPv4 provides is 4.3×10^9, thus global deployment needs to adopt IPv6.

E. Comparison of global and stub network deployments

From previous experiments, we can see that, implementing either of the two deployments will reduce routing table size and update rates significantly.

In terms of scalability, global deployment performs better than stub network deployment. In global deployment, 90.2% of the ASes will have a routing table with less than 3,000 entries (reduced by 99.1% comparing with current routing table size). In contrast, in stub network deployment, the table size is between 12,517 and 21,093 (reduce by 93.3-96.3%). Moreover, global deployment has a smaller BGP update rate. However, in global deployment, there are 25 ASes who will have 30,000-70,000 entries in their routing table. This is because, in global deployment, each multihomed AS should inherit a sub-prefix from each of its providers, and each sub-prefix should be inherited by every customer. If an AS needs to pass a number of provider-customer links until it reach a Tier-1 ISP, it is possible that the AS will inherit a large number of prefixes, and all these prefixes will show up in its routing table. If the AS peers with another AS, all these prefixes will propagate to the other AS's customers without aggregation. Fortunately, the current Internet only has a small number of this kind of ASes, and most of ASes are only several hops away from Tier-1 ISPs. As shown in our experiments, we only has a small number of ASes with more than 30,000 entries. Thus, global deployment is better in improving the Internet scalability.

Even though global deployment achieves a better performance, comparing with stub network deployment, it is harder to deploy. Global deployment requires a whole new arrangement of IP addresses in the Internet. In contrast, stub network deployment only requires changes in the stub networks. Further, global deployment requires a large address space which cannot be provided by IPv4, while the stub network deployment only uses an extra 33 /8 prefixes. Thus, stub network deployment is easier to implement.

V. Conclusion

HAA provides a way to handle the Internet scalability problem. However, it causes robustness problems in the Internet routing. In this paper, we propose a light-weighted routing method RD to solve these problems. RD reacts to access failures fast with small packet loss, and it is isolated from BGP. We prove that RD is robust under any network event. Our experiment with real AS topology shows that HAA with RD improves the scalability significantly.

VI. Acknowledgements

This work is supported by the National Science Foundation under Grant CNS-0917078 and CNS-1036496.

References

[1] Routeview project. In *University of Oregon Advanced Network Technology Center*. http:www.routeviews.org.

[2] T. Bu, L. Gao, and D. Towsley. On characterizing BGP routing table growth. In *In Proceedings of IEEE Global Internet Symposium*, 2004.

[3] D. Chang, R. Govindan, and J. Heidemann. An empirical study of router response to large BGP routing table load. In *in Proc. Internet Measurement Workshop*, pages 203–208. ACM Press, 2002.

[4] K. Fujikawa, K. Ohira, and M. Ohta. A hierarchical automatic address allocation method considering end-to-end multihoming. In *IEICE Tech Rep*, volume 109, 2009. IA2009-56.

[5] K. Fujikawa and M. Ohta. Cooperation of hierarchical automatic locator number allocation protocol HANA and DNS. In *IEICE Tech*, volume 110, 2010. IA2010-45.

[6] L. Gao. On inferring autonomous system relationships in the internet. *IEEE/ACM Transactions on Networking*, 9:733–745, 2000.

[7] R. Gummadi and R. Govindan. Practical routing-layer support for scalable multihoming. In *in IEEE Infocom on Computer Communications*, 2005.

[8] G. Huston. The BGP instability report. http:bgpupdates.potaroo.net.

[9] G. Huston. Analyzing the internet's BGP routing table. *The Internet Protocol Journal*, 4, March 2001.

[10] G. Huston and G. Armitage. Projecting future IPv4 router requirements from trends in dynamic BGP behaviour. In *In Australian Telecommunication Networks and Applications Conference (ATNAC*, 2006.

[11] Y. Song, L. Gao, and F. Kenji. Resilient routing under hierarchical automatic addressing, 2011. http://rio.ecs.umass.edu/ỹsong/RD.pdf.

[12] X. Zhao, B. Zhang, D. Massey, A. Terzis, and L. Zhang. The impact of link failure location on routing dynamics: A formal analysis. In *in Proc. of ACM SIGCOMM Asia Workshop*, 2005.