Towards an Aggregation-aware Internet Routing

Yangyang Wang, Jun Bi, Jianping Wu
Network Research Center, Department of Computer Science of Tsinghua University
Tsinghua National Laboratory for Information Science and Technology (TNList)

***Abstract*—Internet is composed of a large amount of autonomous systems (ASes). Border Gateway Protocol (BGP) is the de facto standard used to connect these ASes and exchange reachability information between them. The global BGP routing table size in default free zone (DFZ) grows fast due to many factors including IP address allocation, multihoming, and traffic engineering, etc. Increasing prefix fragments consume more memory space and computational capacity in network forwarding devices. It has been known that the Internet has a potential routing scalability issue along with large address space (e.g., IPv6) deployment in the future. Route aggregation is a practical approach to reduce route entries. In this paper, we propose an innovation based on BGP, named Aggregation-aware Inter-Domain Routing (AIDR). It takes advantage of the redundant paths to the same destination in the Internet, and takes route aggregation into account in route selection to get more aggregation for forwarding table (FIB). We give a detailed analysis and evaluation on the effect of AIDR using the public BGP traces from RouteViews and RIPE. It shows that AIDR can produce aggregated FIBs of size roughly 20%∼36% of the original routing table size with allowing 2.0 AS path stretch, and 25%∼40% without AS path stretch.**

***Index Terms*—Internet routing, routing scalability, route aggregation.**

I. Introduction

Today's Internet is a huge network comprised of tens of thousands of autonomous systems (ASes). Border Gateway Protocol (BGP) is the standard routing protocol used to exchange network reachability information among the ASes. In recent years, the global BGP routing table size in default free zone (DFZ) grows at super-linear rate. It indicates that future Internet based on IP will face a potential challenge in routing scalability. The routing table size expansion is attributed to the growing number of networks connected to the Internet, unaggregatable IP address allocations, and prefix deaggregation due to multi-homing and traffic engineering [1]. These factors cause many more-specific prefixes to be announced into the global routing table. These prefix fragments also lead to a fast expanding forwarding information base (FIB) that will consume more memory space and computational capacity in highly-efficient packet forwarding devices. It has been recognized that routing scalability issue is one of important problems in future Internet design.

Many efforts have been made to improve Internet routing scalability. One of views believes that routing scalability is one of architecture deficiencies of the Internet, and it is necessary to take architectural changes to conquer these problems. For example, core-edge separation solutions and core-edge elimination solutions [2]. The core-edge separation solutions,such as LISP [3], eFIT [4], sperate edge networks from transit core networks so that the prefixes of edge networks can not be propagated into transit core, and then reduce the routing table size in transit core. In this approach, a mapping between the two address spaces of edge networks and transit core is a necessary component. The core-edge elimination solutions, such as Shim6 [5], ILNP[6], split the semantics of identifier and locator of IP addresses. In this approach, a multihoming network may inherit multiple provider-allocated address spaces that can be aggregated in provider networks. A host of that network will possess multiple IP addresses as locators and use one of them as identifiers to identify host communication ends in transport layer. It also needs a global mapping between identifier and locator. Besides these practical proposals, there are some studies on compact routing [7] that attempt to find theoretical solutions on Internet-like graphs. These approaches take different routing architectures than today's Internet, and their deployment could be a long-term process.

Some other studies design practical methods to routing scalability with few architecture changes. Ballani et al. propose a prefix virtual aggregation (VA) method [8] that divide the address space of RIB and FIB into a number of small pieces and distribute them among the router within an AS. It can be deployed individually by ISPs. Zhang et al. [9] discuss the evolution from intra-domain VA islands to a continuous area of inter-domain VA after merging neighboring VA islands, which finally lead to the of result of core-edge separation. Some studies, such as ORTC [10] and 4-level [11], focus on FIB aggregation, that aggregate route entries while retaining the equal forwarding effect to the original FIB. These FIB aggregation algorithms can be locally deployed on routers without any requirement to modify the Internet architecture.

In this paper, we further exploit the aggregatability of routing tables. As the connections among ASes increase, there may be multiple available paths from one network to the same destination. To leverage the route diversity in route aggregation, we propose a mechanism based on BGP, named Aggregation-aware Inter-Domain Routing (AIDR). It takes route aggregation as one of the considerations in route decision policies. It is possible some routes with AS path stretch are selected as the best routes for the purpose of route aggregation. There is a tradeoff between aggregation and other requirements. In practice, current Internet routing does not always prefer shortest paths due to complicated routing policies among ASes. Measurements [12] indicate that at least 55% of AS paths has one AS hop of inflation. It is reasonable to save the overhead of memory and computation in routers by

978-1-4673-1544-9/12/$31.00 ©2012 IEEE

route aggregation with acceptable AS path stretch, especially in the critical situation of routing table inflation in the future. We also give a detailed evaluation and analysis for AIDR using the BGP routing table traces from RouteViews [13] and RIPE RIS [14]. We estimate the path stretch and the number of available paths under a given stretch limitation at AS-level. The results shows that, on average, more than 10% of AS neighbors for an AS can provide alternative route to prefix with no stretch. It can construct aggregated FIBs of size roughly 25%∼40% of original routing table size with no AS path stretch. We also compare the effect of AIDR with ORTC and 4-level. The results show that AIDR can get more FIB reduction than other two methods.

The rest of the paper is organized as follows. The description of AIDR and related algorithms is presented in Section II. And section III gives the evaluation to AIDR. Section IV is related work, and the last section is a conclusion.

II. Framework and algorithms of AIDR

In this section, we present the basic idea and framework of AIDR. And then, We describe the algorithms of how to choose routes to improve aggregation of forwarding table, and the process of route updates.

A. Overview of AIDR

The basic idea in our aggregation-aware approach is to take aggregation into account in the process of route selection. An example is shown in Fig. 1, where each circle labeled a number represents an AS. AS1 learns two routes for prefix 1.1.0.0/17 with AS paths (2,4,d) and (4,d), and one route for prefix 1.1.128.0/17 with AS path (2,3,d). Of course, AS2 can aggregate 1.1.0.0/17 and 1.1.128.0/17 into a prefix 1.1.0.0/16, and announce it to AS1. However, in practice, AS2 is not ready to aggregate them because of traffic engineering or other factors. In AIDR, AS1 can decide the tradeoff for route aggregation in route selection individually. In this example, the best routes from AS1 to 1.1.0.0/17 and 1.1.128.0/17 are (4,d) and (2,3,d) respectively in terms of shortest AS path. They cannot be aggregated in the local FIB in the AS border routers of AS1 due to different nexthops. If AS1 selects (2,4,d) as the best route to 1.1.0.0/17, these two prefixes have the same AS nexthop, and can be aggregated into one prefix 1.1.0.0/16 in the local FIB of AS1. In this example, there is one AS hops of path stretch. There alos could be no path stretch in other situations.

B. Framework of aggregation-aware routing

In today's Internet routing, BGP instances on AS border routers usually maintain three types of routing tables [15]: import routing tables, denoted as Adj-RIB-in, that records the routes learned from neighboring BGP peers. After applying import policies and route selection rules on Adj-RIB-in, the router work out the best routes to be used in this router, and stores them in a local routing table, denoted as Loc-RIB. The best routes will be disseminated to Adj-RIB-out routing tables according to export policies, and propagated to its BGP neighbors. BGP may filter out the longer IP prefixes in the import policies. Route aggregation may be applied on the routes in Adj-RIB-out before being exported.

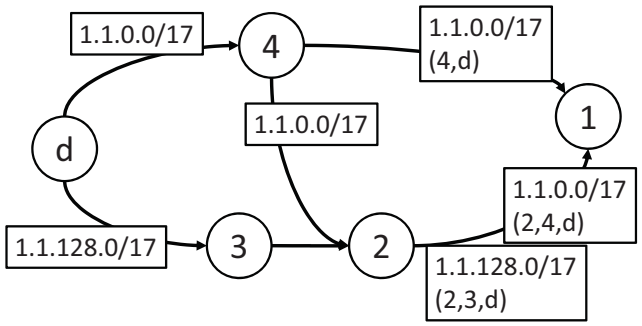

Fig. 1. Example. AS1 prefers the path (2,4,d) to (4,d) to aggregate 1.1.0.0/17 and 1.1.128.0/17

rank	route selection steps
1	highest local pref
2	shortest AS path length **- replaced with aggregation consideration with constraint on AS path stretch**
3	lowest origin value
4	lower MED
5	eBGP-learned over iBGP-learned
6	lowest IGP cost/distance
7	lowest Router ID

TABLE I
AGGREGATION-AWARE ROUTE DECISION STEPS

AIDR improves this process by considering aggregation in the route selection rules. As showed in TableI, we relax the shortest path rule, and take aggregation into account with compromising path stretch. After applying the second aggregation-aware rule, there could be multiple routes left, and they will go through the rest tie-break decision rules based on other BGP path attributes, such as ORIGIN, Multi-exit Discriminator (MED), etc. Current BGP route selection can make decision on each route independently. In AIDR's decision process, one route choice may depend on other route choices to achieve an optimal aggregate state in the FIB. This is because multiple prefixes that are consecutive in address space need to share the same forwarding nexthops to be aggregatable in local routing table.

The Figure 2 describes the flow of AIDR route process. As mentioned previously, Adj-RIB-in stores the routes learned from its neighboring BGP peers. After applying import policies and aggregation-aware rules, the best routes are selected out, and stored in the Loc-RIB. The original FIB is generated based on the Loc-RIB, and an aggregated FIB can be obtained from the original FIB by FIB aggregation. Because AIDR route selection is more complicated than current BGP route selection, it is not cost-effective to recalculate aggregated FIB upon receiving route updates. We will present a method to update aggregated FIB incrementally.

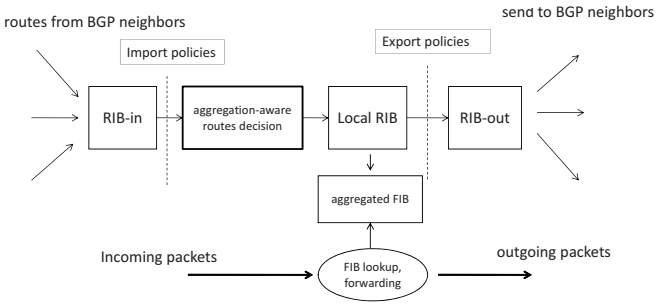

Fig. 2. The flow of AIDR route process

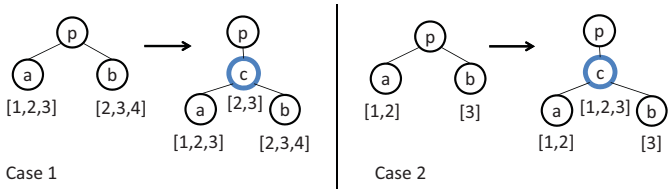

Fig. 3. Horizontal operation

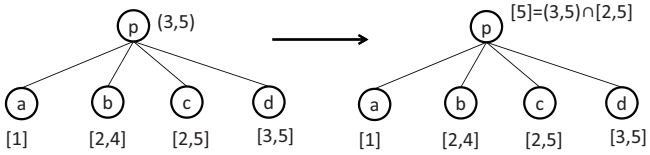

Fig. 4. Vertical operation

C. Algorithm for route aggregation decision

In this section, we describe the algorithm of aggregation-aware route decision. A few routes may be filtered according to the rules, such as local preference, prior to the aggregation-aware rule in the Table I. Then, the remained routes go into the process of aggregation-aware selection.This process firstly check the constraint on AS path stretch, denoted by α here. Only the routes with path stretch less than or equal to α will pass into the following selection. If set $\alpha = 1.0$, that means only the routes of shortest path are permitted. We can set α different values for different routes as we need. After this filtering, we will get a set of routes comprised of specific prefixes with available paths nexthops. The next step is to find one or more selection decisions that can contribute the most aggregatability to the local forwarding table.

This problem can be described as follows. Given a set of prefixes, each prefix has multiple paths associated with it, which means a prefix may be associated with multiple optional nexthops. We define a *selection* as a decision of assigning one nexthop to every prefix. The problem is how to find one or more selections that can lead to an optimal FIB aggregation from all possible selections. The meaning of the term "optimal" depends on practical needs. Here, we refer to the optimal result as the minimum number of entries (e.g., prefixes) in aggregated forwarding tables.

To describe this route selection algorithm, we define some concepts. If the address block of prefix a is contained in the address block of prefix b, we call that b *covers* a. If there is no other prefix that covers a but is covered by b, we call that b *immediately covers* a. In this route selection algorithm, every prefix is represented as a node, and all prefix nodes construct a tree based on the immediately covering relation between prefixes. We assume that the root of the tree is the default route. Each prefix node but root in this tree carries one or more *available nexthops*. We use $AH(a)$ to denote the available nexthops of prefix a. This algorithm has two passes. The first pass performs a post-order traversal of the prefix tree from the bottom up to the root to calculate the *candidate nexthops* for each prefix node. The candidate nexthops are the most prevalent nexthops at the level of every node in the tree. For the leaf nodes, their available nexthops are just the candidate nexthops. We use $CH(a)$ to denote the candidate nexthops of prefix a. In this traversal process, two kinds of operations are executed on the tree: the horizontal operation and vertical operation. The detailed steps are described as follows.

Horizontal operation: This kind of operation is performed on the child prefix nodes $children(p)$ of a parent p. It has two cases. **case 1)** If the address blocks of two child prefixes a and b are consecutive in address space, and they have a common set of nexthops, they will be aggregated into a less-specific prefix node c with a set of candidate nexthops $CH(c) = CH(a) \cap CH(b)$. **case 2)** If the address blocks of two child prefixes a and b are consecutive in address space, but they have no common nexthop, they will be aggregated into a less-specific prefix node c with a set of candidate nexthops $CH(c) = CH(a) \cup CH(b)$. In the horizontal operation, if prefix c is not existing in the tree, a new node c is created in the tree. Then, c is added to $children(p)$, and a and b are moved from $children(p)$ to $children(c)$. Figure 3 shows an example of the two cases of horizontal operation. The numbers in square brackets represent the candidate nexthops of each node. If prefix c is already in the tree and is just p, then don't do horizontal operation on a and b. Repeat the horizontal operation on the set $children(p)$ until there is no updates in $children(p)$ (i.e., there is no two child prefixes are aggregatable).

Vertical operation: The vertical operation happens between parent p and its children. It starts after finishing the horizontal operation on the children of prefix p. Suppose that prefix p has a set of children $c_1, c_2, ..., c_n$. We count the times of appearance for each candidate nexthop of these children to calculate the most prevalent candidate nexthops. It is denoted as $prevalent(CH(children(p)))$, where $CH(children(p)) = CH(c_1) \cup CH(c_2) \cup ... \cup CH(c_n)$. And $CH(p) = AH(p) \cap prevalent(CH(children(p)))$. Figure 4 gives an example. The node p has child nodes a, b, c and d. The prevalent candidate nexthops of the child nodes are 2 and 5, each of which appears two times. And the numbers in the parentheses close to p represent the available nexthops of p. The candidate nexthops of p is $\{2, 5\} \cap \{3, 5\}$

After iterate the two operations at each parent-child level in the tree by post-order from bottom to top, we can obtain

the candidate nexthops (i.e.,route paths) of each prefix. The order of horizontal and vertical operations has impact on the effect of aggregation. It can be proved that *"horizontal-first-vertical-last"* $\geq$ *"vertical-first-horizontal-last"* $>$ *"vertical-only"*. And then, it uses a pre-order traversal of the tree from top to bottom to make the final route selection decision for each prefix and estimate the number of aggregate prefixes. Firstly, we randomly choose the nexthop k that has shortest AS path from the candidate nexthops of the root node. And then, traverse the children of the root node. If k is a member of the candidate nexthops of the visited child, this child prefix can be aggregated into its parent prefix p. And we label this child node with "not-exist" and its route selection is k. If the visited child has no k in its candidate nexthop set, this child cannot be aggregated into p. It will be labeled as "exist", and we choose the nexthop of shortest AS path as its route selection. Iterate this process in the traversal. Finally, each node has a label to indicate its route selection and whether it exists or not in the aggregated forwarding table. We can perform this process from root again to get another different route selection if there exist multiple equally optimal route selections.

D. Process of updates

When an AIDR router receives route updates, it needs to recalculate the full routing table to get an optimized route selection. However, frequent recalculation may cause CPU heavy load. Thus, we take a method to update aggregated FIB incrementally in the intervals of successive full recalculation. We denote the announcement and withdraw for prefix p as $A(p)$ and $W(p)$, and use $fib(p)$ to denote the matched route entry of the prefix p or a prefix covering p in the aggregated FIB. We also assume that there are Adj-RIB-in tables storing the routes from neighboring BGP peers, and a local RIB storing best route selection of AIDR, and a aggregate FIB that is derived and aggregated from the local RIB. The process is described in two classes as follows:

1) **Announcement:** If $A(p)$ is a duplicated update that does not update anything of the best route of p, no need to update RIB and FIB; If $A(p)$ is a new route that $fib(p)$ is empty, then insert p into the local RIB and aggregated FIB; If $fib(p)$ is not empty, and $A(p)$ is from the AS nexthop of $fib(p)$, it will update local RIB and insert p into aggregated FIB when the $A(p)$ has an acceptable AS path stretch, otherwise another best route will be set (such as the one with shortest path) for p in local RIB, and inserted into aggregated FIB. If $fib(p)$ is not empty, and $A(p)$ is from a AS that is not the AS nexthop of $fib(p)$, then ignore this $A(p)$ because it doesn't change the routes in local RIB and aggregated FIB.

2) **Withdraw:** If the $fib(p)$ is empty, ignore this withdraw $W(p)$; Otherwise, select another available route for p from RIB-in tables as the best route, and insert it into RIB and aggregated FIB; If no other routes available for p and the aggregated FIB has a entry for p (i.e., fib(p) is not empty), then delete p from RIB, and insert p to aggregated FIB and set it unroutable, which means packets matched this entry will be discarded. This is like a blacklist.

III. Evaluation and Analysis

In this section, we estimate the feasibility and effect of AIDR using publicly available BGP routing tables.

A. Methodology

We collected BGP routing table snapshots of the date April 22, 2010 from the 8 collectors of RouteViews [13] and 14 collectors of RIPE [14]. And then, we extract the prefixes and AS paths from these routing tables, and ignore the paths containing private AS and AS loops. If the AS set of an AS path contains a single AS, we regard the single AS as the possible last AS hop of the AS path. The BGP traces are just the local RIBs of monitored ASes, not the RIB-in tables of the monitored ASes learned from their neighoring ASes. If we consider a collector as a router, we can regard the collection of the monitored RIBs as the RIB-in tables of this collector. The collectors *'oix-route-views'*, *'route-views2'*, *'route-views3'* from RouteViews, and the collectors *'rrc00'* and *'rrc01'* from RIPE are used in this way. They are denoted by `'rv'`, `'rv2'`, `'rv3'`, `'rrc00'` and `'rrc01'` in the following. The monitored BGP routing tables have no the information of IP nexthop for each route. To get more data samples, we also try to infer the RIB-in tables of some tier-2 and tier-1 ASes. The inferring process is as follows. From every prefix and its AS path in a BGP routing table, we can know the best AS path from each traversed AS hop to this prefix, because each AS announces the best path to its neighboring AS. For example, a route 12.11.8.0/24 with AS path (293,7018,36268) means that AS7018 has the best path (7018,36268) and AS36268 has the best path (36268) for 12.11.8.0/24 in their RIB. Based on all the collected BGP data, we can partly conclude the routing tables of the neighboring ASes of a given AS, which may construct a part of the RIB-in table of the given AS. We use this method to infer the possible RIB-in tables of AS3320, AS5511 and AS7018. In our evaluation, we use AS-level nexthop as an approximate estimation of IP-level nexthop. This may lead to an underestimation to the effectiveness of aggregation because different AS nexthops may share the same IP nexthop [11].

B. Nexthop vs. Path Stretch

In a RIB-in routing table, a prefix may correspond to more than one AS paths that are received from multiple neighboring ASes. We use the number of different AS nexthops to evaluate the diversity of available paths for one prefix. This may underestimate the path diversity because different AS paths may share the same AS nexthop. If denote $l(p)$ as the length of a path p, and denote l_{min} as the shortest length over all paths, then we evaluate each path stretch by $l(p)/l_{min}$. Note that, some paths include AS prepending for traffic engineering. To keep the impact of traffic engineering, we do not eliminate prepending ASes. Figure 5 shows the cumulative distribution function (CDF) curve of the average number of AS nexthop over all AS paths seen from a vantage point, with a growing path stretch constraint. The `y` axes is normalized by the maximum average number of AS nexthops. We can see that

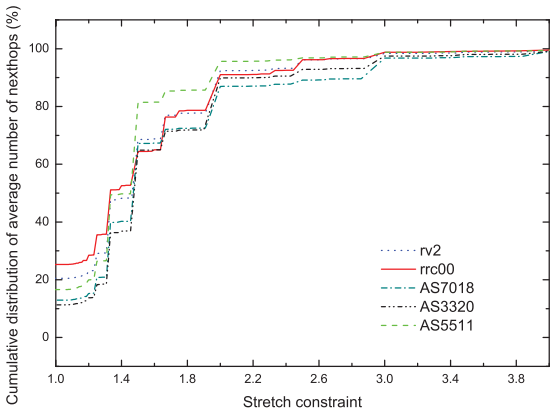

Fig. 5. Cumulative Distribution Function of average number of AS nexthops over all paths

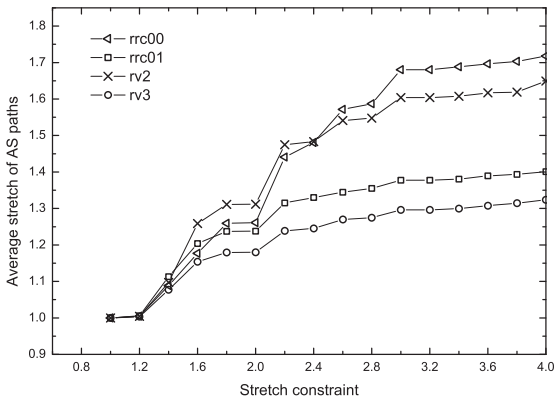

Fig. 7. Average path stretch of AIDR

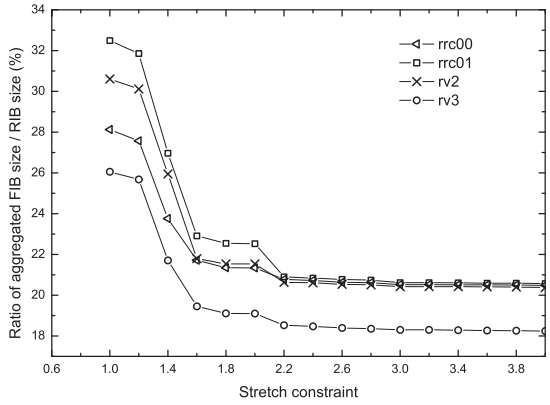

Fig. 6. Table size ratio of AIDR

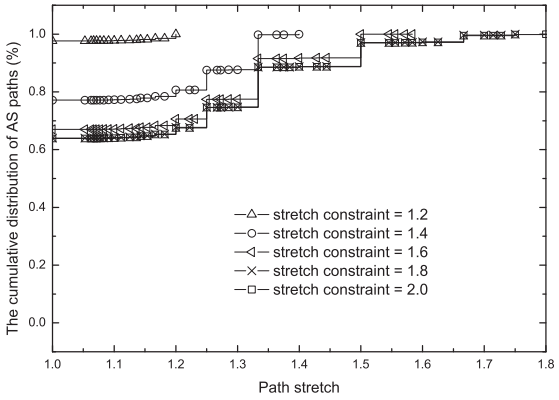

Fig. 8. Cumulative distribution of path stretchs under a give stretch constraint

about more than 10% of the neighboring ASes can provide paths with stretch 1.0 (i.e., no path stretch) for five vantage points, and more than 70% with stretch less than 2.0. In fact, the maximum path stretch is more than 4, but the number of paths with stretch more than 4 is very little, and we ignore them in Figure 5.

C. Effect of route aggregation of AIDR

We evaluate the effect of aggregation based on the BGP routing table of collectors. Here, we consider a collector as a router peering with monitored ASes, although it doesn't transit real traffic in the Internet. There is no business relationship between collectors and monitored ASes. Thus we assume that all available routes are equal in terms of local preference. We compute the ratio of aggregated FIB size of AIDR to the routing table size. Figure 6 shows the variation of table size ratio with path stretch growth. We can see that aggregated FIB size can be reduced to about 26% $\sim$ 33% of routing table size with no AS path stretch. As the stretch constraint grows, the table size ratio can be less than 22%. It is almost near to the minimal ratio when the path stretch constraint is close to 2.2. There is a gain about 10% within the stretch range from 1.0 to 2.2. Figure 7 shows the average path stretch based on the BGP data of rv2, rv3, rrc00 and rrc01. It can be found that the average AS path stretch grows slowly with path stretch. It is about 1.1 under at 1.4 stretch constraint, and is less 1.3 at the 1.8 stretch constraint. This indicate the majority of paths have a smaller stretch than the given threshold of stretch constraint.

To have an insight into the path stretch, we calculate the cumulative distribution of path stretches under a given stretch constraint. The result is shown in Figure 8. Under the stretch constraint 2.0, more than 60% paths are the shortest paths, and more than 80% paths have a stretch less than 1.35. This indicate that the majority of paths have a smaller stretch, although the stretch constraint is large. However, we cannot gain more aggregation after stretch constraint large than 2.2 shown in Figure 6 and Figure 9.

In practice, there are business relationships among ASes that affect the route propagation and traffic delivery. The relationship mainly include provider-customer, peer-peer, and sibling. They may give limitations to the number of available nexthops. For example, An AS learns two routes to a /24 prefix a from its customer c_1 and provider p_1 respectively, and it learns routes to another /24 prefix b from two providers p_1

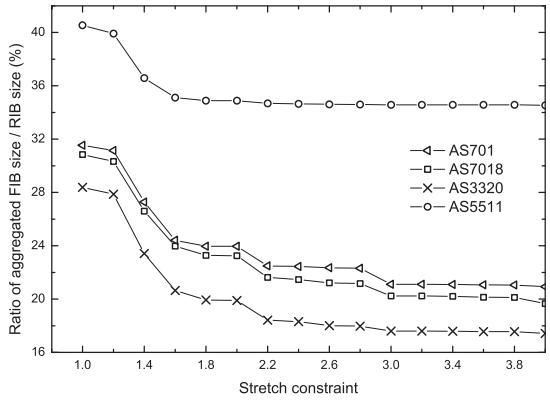

Fig. 9. Table size ratio with route policy constraint

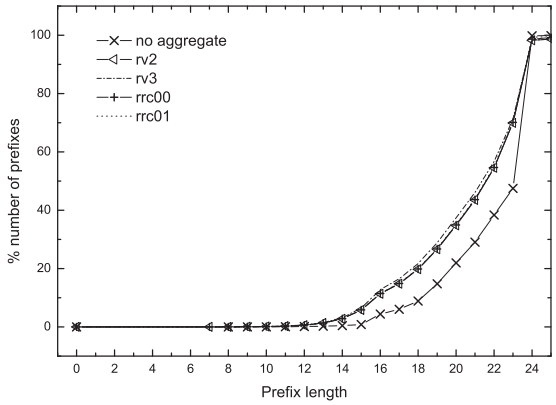

Fig. 10. Prefix length cumulative distribution before and after aggregation

TABLE II
COMPARISON OF FIB AGGREGATION AMONG ORTC, 4-LEVEL, AND AIDR WITH NO PATH STRETCH

FIBs	rv	rv2	rv3	rrc00	rrc01	7018	3320	5511
ORTC	0.32	0.32	0.34	0.30	0.31	0.32	0.35	0.41
4-level	0.31	0.31	0.36	0.27	0.29	0.31	0.32	0.18
AIDR	0.31	0.31	0.26	0.28	0.32	0.31	0.28	0.41

and p_2. Assume a and b are aggregatable in address space, a and b cannot be aggregated in the FIB in the AS. According to the preference rule of customer over provider, a only select the route with nexthop c_1, but b has no route of nexthop c_1.

To evaluate the impact of route policies on the effect of AIDR aggregation, we construct the possible RIB-in routing table and local RIB of AS3320, AS5511 and AS7018. We obtain the AS business relationship of the same date as the collected BGP data, from the map files of UCLA topology project [16]. In routes selection process, we set local preference with customer over peer and peer over provider according to simulate the preference condition [17] in operational practice. From the Figure 9, we can see that, for AS701, AS7018 and AS5511, there are results similar to that shown in Figure 6. The minimal table size ratio can be less 22%, and there is about 10% ratio reduction within the stretch range from 1.0 to 2.2. But the ratio variation of AS5511 is small. This may be caused by the incompleteness in the data about AS5511.

We further compare the prefix length distribution before and after aggregation. The cumulative distribution is shown in Figure 10. It has a notable growth of the percentage of prefixes with length from 14 to 24. The number of prefixes shorter than /22 has a growth of about 15%, and that shorter than /23 prefix has a growth more than 20%. The prefixes of length longer than /24 are ignored in this figure because they are very few.

We implemented the FIB aggregation algorithms of ORTC and 4-level, and applied them to a few FIBs to calculate the ratio of aggregated FIB size to original FIB size. We compare the resulting ratios to that of the AIDR aggregation with no path stretch . The results is shown in the Table II. For the route collector rv, rv2, rv3, rrc00 and rrc01, We use the routes of shortest path to construct their FIBs. And we also get the FIBs of AS7018, AS3320 and AS5511. The results show that 4-level and AIDR have a smaller table size ratio than ORTC. And in some cases, 4-level has a smaller ratio than AIDR. This is because 4-level introduces extra routable address space that fills up the space between prefixes fragments, while ORTC and AIDR have not extra routable space. AIDR get aggregation effect by selecting the proper nexthops from multiple options. If allowing a few path stretch, AIDR can get more aggregation than ORTC and 4-level as shown in previous paragraphs.

D. Updates process

Updates could make an aggregate prefix become deaggregated when a sub-prefix covered by this aggregate prefix is withdrawed. To evaluate the impact of update to route aggregation, we collect one BGP routing table snapshot and the updates in the six hours after the snapshot from the collector `rv2`. It contains 3048073 announcements and 317504 withdraws. Then, we compute the optimal route decision with no limitation on path stretch based on the BGP routing table snapshot. It can generate an aggregated FIB with 66643 prefixes. We handle these updates with incremental method described in previous sections. We try to overestimate the growth by inserting a new entries for all the updates that are from the AS same to the AS nexthop of the corresponding entries in the aggregated FIB. Because this kind of updates may cause the entry changes in aggregated FIB. In our experiment, the aggregated FIB size grows slowly, and it has 84781 entries after handling all updates. The announcements are related to 78057 prefixes, and withdraws are related to only 11720 prefixes in the aggregated FIB. In the related announcements, about 99% (i.e.,3029213/3047677) announcements have different AS nexthop with the corresponding routes in the updated aggregated FIB and can be ignored, in which about 50% (i.e.,1525241/3047677) announcements are related to the sub-prefixes covered by one of prefixes in the updated aggregated FIB. Although announcements related to a entry in aggregated FIB may cause corresponding aggregate route to change, but

most of them are from the AS different from the AS nexthop of the corresponding entry in aggregated FIB, and can be ignored because they don't change the corresponding best route in local RIB and aggregated FIB. But If an announcement is from the AS that is the AS nexthop of the corresponding entry in the aggregated FIB, the announcement has a higher probability to cause change to the route entry.

IV. Related Work

The earlier work [18][19] noticed and measured the growth of BGP routing table size. Bu et al. [18] classify the contributions to BGP routing tables growth, and found that address fragmentation contributes more than 75% to the routing table size, and load balancing gives the fastest growing contribution in the time of measurement. Meng et al. [19] analyzed the impact of IPv4 address block allocation on BGP routing table evolution, and detailedly categories the operational practices of ISPs advertising more-specific prefixes. These measurements also indicate the potential aggregatability in routing table. Draves et al. [10] done a an earlier work on the aggregation of IP forwarding table. It presents a method ORTC that can achieves an optimal aggregation with no extra routable space. It can generate fowarding table of about 60% of the original size in that time of experiments. These work [11][20][21] proposed other method used to forwarding table aggregation. They may introduce extra routable space in aggregation. Zhao et al. [11] systematically evaluate the effect of different level of aggregation. Li et al. [22] present a method of optimal aggregation in the situation of each prefix having multiple selectable nexthop. We improve this method and use it in inter-domain routing. Different from it, we propose an improvement to routing control plane. Atom-based routing [23] aggregates prefixes by policy atoms. A policy atom is a cluster of prefixes that always share the same AS path. Atom-based routing needs cooperation among ASes to tunnel packets, and to spread the mapping information between atom id and destination prefixes. Ballani et al. proposed virtual aggregation (VA) [8]. It divides the whole IP address space into several block called virtual prefix, and distribute them into many routers named aggregation point routers (APRs). APRs has the specific routing table of given virtual prefixes. IP packets firstly are forwarded to a VPR, and that VPR forwards them to the proper export with tunneling. VA can be deployed in ISP network independently, and reduce the forwarding table size in the ISP network. Inter-AS VA can reduce global routing table, but it also require coordination among ASes, and a global mapping system.

V. Conclusion

As the Internet grows, the connectivity among ASes becomes denser than before. One AS may have multiple available AS paths to the same destination through different AS neighbors. In this paper, we propose an aggregation-aware routing scheme. It leverages the path diversity, and takes the prefix aggregation into account in the BGP best route selection. We evaluate the method using public BGP routing tables. We find that averagely more than 10% neighboring ASes can provide alternative routes with no path stretch, more than 70% neighbors can provide alternative routes within 2.0 AS path stretch. In our case study, the aggregated FIB size can be 25%$\sim$40% of original routing table size with no AS path stretch, and 20%$\sim$36% of original size within 2.0 path stretch constraint. And the average path stretch is smaller under a given stretch constraint. These results indicate the potential aggregatability with a little path stretch, and it also implies that we cannot get much more aggregatability by largely relaxing path stretch. In future work, we will enhance the aggregation effect by combining other methods, and study the efficient method of updating aggregated FIBs.

References

[1] D. Meyer, L. Zhang, and K. Fall, "Report from the iab workshop on routing and addressing," in *RFC4984*, 2007.

[2] D. Jen, M. Meisel, H. Yan, D. Massey, L. Wang, B. Zhang, and L. Zhang, "Towards a new internet routing architecture: Arguments for separating edges from transit core," in *Proc. HotNets-VII*, 2008.

[3] D. Farinacci, V. Fuller, D. Meyer, and D. Levis, "Locator/id separation protocol (lisp)," in *Internet Draft, draft-farinacci-lisp-12.txt*, 2009.

[4] D. Massey, L. Wang, B. Zhang, and L. Zhang, "A scalable routing system design for future internet," in *IPv6 workshop with SIGCOMM 2007*, 2007.

[5] E. Nordmark and M. Bagnulo, "Shim6: Level 3 multihoming shim protocol for ipv6," in *RFC5533*, 2009.

[6] R. J. Atkinson, S. N. Bhatti, and S. Hailes, "Evolving the internet architecture through naming." *IEEE Journal on Selected Areas in Communications*, pp. 1319–1325, 2010.

[7] D. V. Krioukov, K. C. Claffy, K. R. Fall, and A. Brady, "On compact routing for the internet," *CoRR*, vol. abs/0708.2309, 2007.

[8] T. C. Hitesh Ballani, Paul Francis and J. Wang, "Making routers last longer with viaggre," in *USENIX NSDI 2009, Boston, MA*, April 2009.

[9] B. Zhang and L. Zhang, "Evolution towards global routing scalability," in *Internet Draft*, 2009.

[10] R. Draves, C. King, V. Srinivasan, and B. Zill, "Constructing optimal ip routing tables," in *INFOCOM*, 1999, pp. 88–97.

[11] X. Zhao, Y. Liu, L. Wang, and B. Zhang, "On the aggregatability of router forwarding tables," in *INFOCOM*, 2010, pp. 848–856.

[12] L. Gao and F. Wang, "The extent of as path inflation by routing policies," in *IEEE Global Internet Symposium*, 2002.

[13] "Routeviews project." [Online]. Available: http://www.routeviews.org/

[14] "Ris raw data of ripe ncc." [Online]. Available: http://www.ripe.net/data-tools/stats/ris/ris-raw-data

[15] Y. Rekhter, T. Li, and S. Hares, "A border gateway protocol 4 (bgp-4)," in *RFC4271*, 2006.

[16] "Internet topology collection of ucla." [Online]. Available: http://irl.cs.ucla.edu/topology/

[17] L. Gao and J. Rexford, "Stable internet routing without global coordination," *IEEE/ACM Trans. Netw.*, vol. 9, no. 6, pp. 681–692, 2001.

[18] T. Bu, L. Gao, and D. F. Towsley, "On characterizing bgp routing table growth," *Computer Networks*, vol. 45, no. 1, pp. 45–54, 2004.

[19] X. Meng, Z. Xu, B. Zhang, G. Huston, S. Lu, and L. Zhang, "Ipv4 address allocation and the bgp routing table evolution," *Computer Communication Review*, vol. 35, no. 1, pp. 71–80, 2004.

[20] B. Zhang, L. Wang, and L. Zhang, "Fib aggregation, draft-zhang-fibaggregation-02.txt," in *Internet Draft*. [Online]. Available: http://tools.ietf.org/html/draft-zhang-fibaggregation-02

[21] W. Herrin, "Opportunistic topological aggregation in the rib-fib calculation?" 2008. [Online]. Available: http://www.ops.ietf.org/lists/rrg/2008/threads.html#01880

[22] Q. Li, D. Wang, M. Xu, and J. Yang, "On the scalability of router forwarding tables: Nexthop-selectable fib aggregation," in *INFOCOM*, 2011, pp. 321–325.

[23] P. Verkaik, A. Broido, Y. Hyun, and kc claffy, "Atom-based routing," in *Presentation*, 2003. [Online]. Available: http://www.caida.org/publications/ presentations/2003/internetworking03/atoms.internetworking03.pdf