

Characterization and Design of Effective BGP AS-PATH Prepending

Ying Zhang, Mallik Tatipamula
Ericsson Research

Abstract—The AS path prepending approach in BGP is commonly used to perform inter-domain traffic engineering, such as inbound traffic load-balancing for multi-homed ASes. It artificially increases the length of the AS level path in BGP announcements by inserting its local AS number multiple times into outgoing EBGp announcement messages. In this work, we first present a comprehensive study on the characterization of Internet routing AS path prepending. We further propose an algorithm for computing the optimal padding strategies given multiple neighboring links. Our method considers the impact of AS relationship based local policies on ASPP’s effectiveness. The algorithm can be used for three objectives, *i.e.*, traffic load balancing, backup route provisioning, and bypassing a specific AS for security purposes, *e.g.*, avoiding information censorship. We demonstrate the accuracy and effectiveness of our approach using real BGP data and traffic data from Abilene networks.

Index Terms—Inter-domain Routing; BGP

I. INTRODUCTION

Traffic engineering (TE) is used by network operators to control the distribution of traffic in face of network condition changes. The common practices evolve adjusting the configuration of the routing protocols running on their routers. A large body of previous work focuses on intra-domain traffic engineering, relying predominantly on Interior Gateway Protocols (IGPs), such as OSPF, IS-IS, and MPLS, which control the flow of traffic within a single Autonomous System (AS). In the inter-domain, traffic engineering still relies on operators to make manual changes in the routing policies, without a good understanding of the impact on other domains [1].

Although the inter-domain TE is still in the trial-and-error fashion, it is widely used by ISP operators to balance the load via influencing the BGP routing decision process. The second step in BGP route selection is preferring the route with a shorter AS path length [2]. Thus, one way to influence BGP route selection in the Internet is the extension of the AS path attribute artificially: AS-Path prepending (ASPP), or AS-Path padding. In the normal case, each AS only adds its own AS number once to the AS path. In ASPP, it adds several instances of its own AS number to the AS path to increase the AS-Path length, making this route less attractive to its upstream ASes. More specifically, extra AS-numbers are inserted (prepended or padded) at the beginning of AS-path, just after the local AS-number. By prepending its the local ASN multiple times, ASes can make advertised AS paths appear artificially longer. We name each AS-number appearance in the path a padding instance. An AS may prepend different times for routes through different neighbors.

ASPP is a common practice for local load balancing and for backup route provisioning. For the purpose of load balancing, operators increase the instances of padding at an advertised route in a trial-and-error fashion until the AS-Path length is sufficient to reduce the load of that ingress link to the expected value. For the purpose of provisioning backup routes, the degree of padding is usually large enough so that the backup route will not be adopted as best route unless if there is a failure in other primary routes. Previous measurement study showed that 32% of the routes in the AT&T network have some form of prepending [1]. This observation indicates that ASPP has a significant impact on the current Internet.

Recently there are a number of proposals on general inter-domain TE techniques [1], [3], [4], [5], [6]. Among them only very few studies looked into the ASPP strategy and impact [7], [8], [9], [10]. They mainly focused on the ASPP strategies and its impact on the local AS’s perspective. The strategies are lack of estimation on global impact. Although ASPP has been practised in the Internet for a long time, there has been no systematic study on the performance of this method.

Our objective is to investigate the ASPP behavior on the Internet thoroughly and systematically, and to understand its potential and limitations. We motivate our research through a recent routing anomaly instance caused by inappropriate ASPP behavior. On Mar. 22nd, 2011, traffic to Facebook (AS32924) from most US based ISPs, including AT&T, was routed through China (China telecom AS4134) and Korea (SK Telecom AS9318) [11]. Such abnormal route results in severe long delay due to the long geographic distance it traverses. More seriously people are concerned about the information censorship issue given it traverses through China. We investigate this instance and found that normally Facebook announces the route to its major prefix *69.171.255.0/24* with 5 prepended ASNs, *e.g.*, *7018 3356 32934 32934 32934 32934 32934*. In this instance, a shorter AS path is announced via the Korean ISP AS9318, *i.e.*, *(7018 4134 9318 32934 32934 32934)*, with only 3 padding Facebook’s ASN 32924. The reason can be either Facebook’s misconfiguration or other ASes purposely removing the prepended ASNs 32934. Although the actual cause is still not known, this instance motivates our work to conduct a systematic study on the ASPP behavior and its impact on the global routing system.

We summarize the contribution of this paper below:

- We present a comprehensive characterization of ASPP behavior on the Internet from the public BGP data over a long duration. We find that on average 12% of the routes

Hop	Delay	IP	ASN
1	1 ms	192.168.1.1	
2	41 ms	70.130.143.24	AS7132
3	41 ms	151.164.14.131	AS7132
4	41 ms	151.164.102.106	AS7132
5	131 ms	12.123.30.133	AS7018
6	131 ms	218.30.54.169	AS4134
7	132 ms	202.97.50.37	AS4134
8	137 ms	202.97.49.206	AS4134
9	224 ms	218.30.54.78	AS9318
11	224 ms	198.32.176.71	AS9318
12	245 ms	74.119.77.128	AS32934
13	248 ms	204.15.20.51	AS32934
14	249 ms	69.171.224.39	AS32934

TABLE I
TRACEROUTE FROM US TO FACEBOOK DURING THE INSTANCE

have some amount of ASPP today and this indicates that ASPP has a significant impact on the current Internet routing structure. On the other hand, ISP's topological location has impact on the amount of ASPP observed. Large ISPs, e.g. tier-1 ISPs are more likely to observe ASPP. We identify that ASPP usage is increased largely over time from 2006 to 2011.

- We propose three algorithms that determines the optimal padding for an advertised route through each neighbors of the target network. The algorithms compute the best padding vector for different TE objectives. Each element in the vector represents the number of padding instances for each neighbor. Previous work on computing optimal padding vector [7], [10], [9] all ignore the policy induced preference on route selections based on AS business relationships. For instance, an AS usually prefers to use routes learned from its customers than the one learned from its peers or providers. Our algorithm takes AS relationship into consideration, which is more realistic and thus more accurate in predicting route selection. The validation shows that our prediction can achieve on average 85% accuracy. On the other hand, existing work only considers a single objective, *i.e.*, load-balancing. We extend the algorithms to cope with three different objectives: performing load-balancing, provisioning back-up routes, and bypassing a particular AS.

The outline of the paper is as follows. The details of the Facebook routing anomaly is described in Section II. In Section III, we describe the AS path prepending approach and present analysis methodology. The load-balancing ASPP algorithm is presented in Section IV. We present our measurement characterization results in Section V as well as the validation in Section VI. Related work is given in Section VII and Section VIII concludes this paper with some future work.

II. THE FACEBOOK ROUTING ANOMALY INSTANCE

A recent routing anomaly has been reported that AT&T was routing traffic to Facebook through a Chinese network (China telecom AS4134). Motivated by this blog, we conducted more analysis to investigate the cause of this anomaly instance [11].

We first gather the routing tables and updates from route monitors in RouteView [12] and RIPE [13] on Mar. 22nd, 2011. By looking into all the updates associated with any prefixes announced by Facebook (AS32934), we observed that the anomalous route, (4134 9318 32934 32934 32934), appeared at 7:15:02 GMT and is adopted by almost all large ISPs. For instance, we observed that AT&T (AS7018) changes its route to Facebook via (7018 4134 9318 32934 32934 32934). The problem is not limited to AT&T, NTT, another Tier-1 AS, chose the same route 2914 4134 9318 32934 32934 32934). However, among all ten prefixes announced by Facebook, only two prefixes, 69.171.224.0/20 and 69.171.255.0/24, are affected. Using Planetlab based traceroute experiments, we found that most of the Facebook front-end web servers are in these two prefixes. This cross-ocean detour is further verified by traceroute shown in Table I, which is posted online by an end-user in AT&T's customer network [11]. It is shown that the data path is consistent with the BGP routing path, experiencing longer delay than usual.

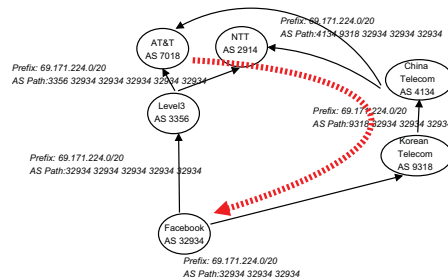


Fig. 1. Facebook routing anomaly instance at 7:15am Mar. 22nd

To understand why this anomalous route is chosen, we seek back to the old route used by AT&T from the routing table snapshot before the change. The BGP level changes are shown in Figure 1. The red dash line shows the flow of the traffic. Interestingly, the old route contains 7 hops, 7018 3356 32934 32934 32934 32934 32934, which is indeed worse than the anomalous route through China. We further examined the routing tables from previous five days and the next five days after the anomaly. All stable routing tables contain this 7-hop route, which indicates that this is the normal route to reach Facebook from most of the Internet.

To summarize, in the normal case AT&T traverses through Level3 (AS3356) to reach Facebook directly from a 6-hop route. Please note that among the 6 ASNs in the route, 5 of them are the Facebook's AS number 32934. At 7:15 on Mar. 22nd, a shorter route is announced from Korean ISP SK Telecom and then through China Telecom with 5 hops. The 5-hop route only contains 3 Facebook's ASN 32934. From the traceroute we observe that the 5-hop route results in much longer delay, thus considered as an instance of routing anomaly. There are many likely causes. The first likely cause is that Facebook purposely announces a shorter route to Korean ISP for traffic engineering by only prepending its ASN twice in the route. This is the most straightforward explanation since AS prepending is commonly used to influence other ASes'

routing decision. The second likely cause is the Korean ISP AS9318. In this case, Facebook still announces a route with 5 of its own ASNs. AS9318 removes two of them and sends to its peer China Telecom. China Telecom just prepends its own ASN 4143 and then announces to the rest of the world. The third possibility is that China Telecom modifies the route directly by removing two of Facebook’s ASNs. The last two cases are considered to be *prefix hijacking* attacks. However, from most monitoring vantage points in US, it is hard to determine which one is the actual cause.

III. ANALYSIS METHODOLOGY ON BGP AS PATH PREPENDING

In this section we first review the BGP background and specifically focus on the AS path prepending (ASPP) method. We then describe our analysis methodology on the ASPP characterization. The characterization analysis helps understand the current practice of ASPP on the Internet.

A. BGP Traffic Engineering and AS path prepending

BGP is the de-facto inter-domain routing protocol for the current Internet. It is used for adjacent routers or ASes to disseminate routing information across the AS border. Each BGP announcement contains a set of attributes. One of the important mandatory attributes is the *AS-PATH*, recording the sequence of ASes through which the message has passed. As an announcement passes, each AS adds (prepends) its own AS number to the front of the *AS-PATH* attribute. BGP is based on distance vector algorithm, meaning that the route with shorter paths is generally preferred.

BGP is also a policy-based routing protocol. The network operators can configure BGP in certain ways to influence the route selection both locally and globally. AS path prepending (ASPP) is one of these traffic engineering approaches. Instead of prepending its ASN once to the path, an AS adds its own AS number multiple times to artificially increase the length of the AS path. Assume a BGP announcement for prefix p has an *AS-PATH* $\{AS_1, AS_2 * \dots AS_k\}$, where $*$ stands for one more occurrence of AS_2 . The longer the AS path is announced to the EBGP neighbor, the less likely the route will be adopted as the best route by other ASes, indicating that the less incoming traffic will be received from that neighbor. When manipulating AS paths, the only valid AS number to prepend is the AS number of the sender. Prepending any other AS number is considered as misbehaving. The AS which prepends its AS number multiple times (> 1) into the AS path is called *prepending AS* or *padding AS*. The ASPP can be classified to two types, source prepending and intermediary prepending, based on the location of the prepending AS. Source prepending is referred to the case that padding is performed by the origin ASes or the owner of the prefixes, while the intermediary prepending is performed by other non-origin ASes along the path.

Through ASPP, an AS could influence the route selection process and thus affect the distribution of traffic flowing into it. When multiple routes are available, BGP follows the decision process in Table II to select the best one. Therefore, the

- | |
|---|
| <ol style="list-style-type: none"> 1. Ignore if the next hop is unreachable 2. Highest local preference 3. Shortest AS path 4. Lowest origin type 5. Lowest Multiple-Exit-Discriminator (MED) value among routes from the same AS 6. eBGP learned route over iBGP learned route 7. Lowest IGP cost (hot-potato) 8. Lowest router ID |
|---|

TABLE II
BGP DECISION PROCESS

<pre>router bgp 65001 neighbor 10.1.0.2 remote-as 65200 neighbor 10.1.0.2 description Backup ISP neighbor 10.1.0.2 route-map prepend out route-map prepend permit 10 set as-path prepend 65001 65001 65001</pre>
--

TABLE III
CISCO CONFIGURATION ON AS PATH PREPENDING

effectiveness of any AS-Path prepending is limited by the use of local routing policies expressed through the Local Preference attribute, which has a higher priority in the decision process. A common local preference policy is that, for the same destination prefix, an AS prefers to send traffic through a customer link than a peering link, and it prefers to use a peering link than a provider link [14]. This is because an AS does not need to pay for the traffic going through its customer link but has to pay for traffic traversing the provider links. One way to express this policy is using local preference attribute in the announcement. Another way to realize the policy is using the selective announcements. It is recommended that an AS should not announce routes learned from its providers/peers to other providers and peers, in order to guarantee it does not provide transit service between its providers or its peers. It is often known as the “valley-free” policy [14].

We briefly introduce how AS-path prepending can be configured on the routers. Table III shows the configuration on Cisco routers using route-map based on per-neighbor outbound filter. The actual prepending is specified within the route-map with the set as-path prepend command. Cisco IOS supports inbound and outbound AS-path prepending on EBGP sessions. AS-path prepending can be only applied on EBGP sessions. Outbound AS-path prepending can be used as the last-resort mechanism to influence global BGP routing policies where all other methods (Multi-exit Discriminator or Local preference manipulation) is not supported by upstream ISPs. It is also more powerful than MED and LocalPref when there is a large difference in upstream ISPs’ connectivity to the remaining Internet, due to its global impact. An AS can also ask other ASes to do prepending for it through the community attribute. We plan to study this option in the future work.

B. Characterization of AS path prepending behavior

To fully understand the practice of ASPP on today’s Internet, we conduct a set of analysis on BGP routing data over a

long period for ASPP characterization.

1) *Data source and preprocessing*: Our study draws on BGP update messages from the publicly-available RouteViews [12] and RIPE NCC [13] servers. These public servers collect update messages by establishing eBGP sessions with routers in participating ASes. The logs contain the best route from all the peering routers. Our study uses routing table and update data for five years from 2006 to 2011 because the set of monitors are consistent over this period. We start with an initial BGP routing table and apply the stream of update messages to construct a view of the routing table at each point in time.

We combine routing table files in the most recent three months to identify the AS level links. We then infer the AS relationship using the topology graph. We first generate graphs using Gao’s algorithm with only Tier-1 peering links as the initial input. We did the same calculation using CAIDA’s algorithm. Then we take the set of relationship pairs upon which both graphs agree. We take the common set as the new initial input to re-run Gao’s algorithm to generate our topology graph. Although obtaining a precise AS topology is difficult, we attempt to improve its accuracy to the maximum extent.

2) *Characterization of prepending AS’s behavior*: An AS may perform prepending for different purposes. For instance, it can be used to shape the incoming traffic from the upstream providers, for the purpose of performance improvement or reducing total cost. While load balancing is a common objective for ASPP, an AS can choose to let a large portion of the incoming traffic to use a specific transit AS, which is more reliable or has higher bandwidth availability. It can also be used to adapt the distribution of traffic flows within the ISP. Given the variety of objectives, different ASes may perform prepending in different ways. We characterize its behavior to answer the following questions.

- What is the fraction of routes associated with prepending behavior? How does the location of the monitoring vantage point affect the observed fraction?
- What are the ASes that perform prepending? Where are they in the Internet topology, core or edge? Where are they in the AS path, origin ASes or intermediate ASes?
- How do they prepend? How many instances do they add? Is the policy consistent across all prefixes through the same AS?

From the routing table and update data, we first collect all the routes with prepending AS path. For each route, we extract the information of prefix, origin AS, and ASes that are prepending. Then from the data, we further identify all other routes, regardless of having prepending AS-path or not, sharing the same prefix, or origin AS, or prepending ASes. Then we perform correlations from multiple dimensions, *e.g.*, correlating routes with shared prefix, shared origin AS, and shared prepending AS.

3) *Characterization of dynamic trends*: Another metric of characterizing the ASPP usage is its growth and dynamics over time. Trend analysis depicts the involvement of multi-homing and traffic engineering behavior on the global Internet. It provides insights on how BGP is used by the operators for

```

1. Compute all uphill paths from  $t$ 's providers set  $N$  to all ASes,
   path is stored in array  $PD$  and distance is stored in  $D$ 
2. function  $update\_dist\_vec(k,D,PD)$ 
 $\#D = \{d_{k,j}\}$ : distance from  $AS_k$  to nei  $j$ ,  $PD$ : array of paths for  $D$ 
   if  $\exists j, D_{k,j} < \infty$  # choose customers' path
       store path in  $PD$ , return  $D$ ;
   else # find paths through peer or provider
       list all  $k$ 's peers in set  $P = (p_1, \dots, p_g)$ 
       computed(k,D,P)
   if  $\exists j \& j \notin N^1, D_{k,j} < \infty$  # choose peers' path
       store path in  $PD$ , return  $D$ ;
   else # choose provider's path
       list all  $k$ 's providers in set  $P = (p_1, \dots, p_g)$ 
       for all  $p \in P$ 
           update_dist_vec(p,D,PD) # update each provider recursively
       computed(k,D,P) # update D
3. function  $computeD(k,D,P) \# P = (p_1, \dots, p_g)$ 
   for all  $j = 1$  to  $m$ 
        $d_{k,j} = \min(d_{p_1,j}, d_{p_2,j}, \dots, d_{p_g,j}) + 1$ 

```

Fig. 2. Algorithm to compute comparable paths for all nodes to t

various TE purposes. In particular, we investigate its usage popularity and impact. For each metric, we compute the average amongst all prefixes and ASes every day and examine the distributions over time.

4) *Characterization of Internet-wide impact*: AS path prepending essentially is a method to selectively adjust the preference of links between the prepending AS and its neighbors, so as to influence the amount of traffic passing through these links. ASPP can result in both local and global impact. Locally, a multi-homed AS can adjust the traffic on incoming links from different providers based on various factors such as pricing or available capacity. Thus, local AS can perform traffic engineering efficiently. Globally, the prepending AS can influence the best route chosen by other ASes, indirectly controlling which path other ASes would use to send traffic to itself. For instance, it can influence other ASes not to choose the paths traversing a particular AS. On the other hand, ASPP may increase the total amount of inter-AS resource consumption on the Internet since traffic no longer follows the shortest AS level paths.

We study the Internet-wide impact both at the prefix level and at the AS level. For every announcement with prepending, we first identify all ASes padding their ASNs more than once. For each prepending AS A_{aspp} , we extract all its m providers $P_i (i = 1 \dots m)$ from the AS-level topology pre-computed in Section III-B1. Combining routes observed from all monitors, we classify all other ASes into m groups based on their last hop to reach the destination, P_i . By computing the cardinality of each group, we can infer the impact of this ASPP behavior, *i.e.*, balancing the load, or preferring a particular transit AS, or creating a backup route, *etc.*

IV. GLOBAL LOAD-BALANCING ALGORITHM

AS-Path prepending is a common practice for traffic distribution shaping across multiple providers. The operators need to determine how many ASNs to pad for each ingress link. The set of padding instances for all links is called the padding vector. Each prefix has its own padding vector. Today, most

operators determine the value in a trial-and-error manner. Previous work studied efficient algorithms to calculate the optimal padding vector to balance the load among N ingress links [7]. These work formulated it as an integer programming problem on a graph and developed polynomial algorithms or heuristics to solve it. Previous work only focuses on balancing the ingress traffic locally. It does not consider the global impact of an ASPP decision. Moreover, it does not consider the impact of realistic local preference policies on ASPP's effectiveness. It assumes that local preference is not even used [9], which is an unrealistic assumption.

Compared to existing work, our algorithm has three contributions. First, we propose a mechanism for calculating optimal padding vectors taken into consideration the local policy preference based on AS commercial relationships. Instead of focusing on the impact of traffic distribution on ingress links, we predict the route adaptation on all other ASes in the Internet. Thus, we are able to estimate the global impact of any padding action. Finally, we extend the objectives of ASPP beyond load-balancing. We investigate three objectives, balancing the load, provisioning a backup route, and bypassing a specific AS. Bypassing a specific AS is needed when an origin AS does not want a particular AS along the path for traffic destined to it. For instance, Facebook may not want its posts to traverse China due to information censorship. In the following, we present our algorithms for the above three purposes.

A. Computing sets of comparable routes

Let I be the set of all ASes on the Internet. Let's consider a specific target AS t , which will announce a prefix p to its neighbors $N = \{n_i\}$, where $N \in I$. N includes both its peers denoted by N^1 and its providers by N^2 . Each AS in I will receive several routes for prefix p and choose the best one according to the AS path length. By using prepending, the target AS t can influence the way in which other ASes will choose the route to reach p . The prepending strategy AS t apply is denoted as $\Psi = \{\psi_i\}$, inserting its identifier ψ_i times in the AS-path of the route announced to neighbor n_i . The problem is how to determine the vector Ψ given an objective function.

One naive approach to compute Ψ is to enumerate all possible combinations and simulate the route selection for each combination. However, it is not scalable when the set of neighbors is large. Another method is to compute and store all possible routes from all other ASes to the target AS. But then when applying different ψ_i values, we cannot just simply count the length of an AS path because the AS relationship may affect the route selection. For instance, from AS k to t , route r_1 has 3 hops and r_2 has 4 hops, through different neighbors. By padding 2 ASNs to r_1 , r_2 may not be preferred, because r_2 may go through a provider link while r_1 is through a customer link. In this example, r_1 is always preferred over r_2 no matter how the padding is done. In this case, we refer r_1 and r_2 to be *incomparable* with respect to the impact of ASPP. This example inspires us that not all the routes are

```

1. Assignment single-path ASes
  for each  $AS_i \in I$ 
    if  $D_{i,\bar{j}} < \infty \&\& \forall j, j \neq \bar{j}, D_{ij} = \infty$ 
      assign  $AS_i$  to partition  $S_{\bar{j}}$ 
      remove  $AS_i$  from  $I$ , deduct traffic from  $\lambda_{\bar{j}}$ 

2. function partition( $I, S, D, PD, \Psi, \Lambda$ ): #I: set of ASes; S: partition of I
  # $\Psi = \{\psi_1 \dots \psi_m\}$ : padding vector;  $\Lambda$ : max capacity vector
  for each  $\psi \in \Psi$ 
     $\psi = \psi + 1$  #each time  $\psi$  increases by 1
     $benefit_i = compute\_benefit(I, S, D, \Psi, \Lambda)$ 
     $\psi = \psi - 1$  # recovers  $\psi$ 
    choose  $i$  that with maximum  $benefit_i$ ,
     $\psi = \psi + 1$  # changes  $\psi$  with maximum benefit
     $S = update\_S(I, S, D, \Psi)$ 
  if iter > ITER_MAX
    return;
  if  $max_i(benefit_i) = 0$ 
    return;
  partition( $I, S, D, \Psi, \Lambda$ )

function compute_benefit( $I, S, D, \Psi, \Lambda$ )
   $\hat{S} = update\_S(I, S, D, \Psi)$ 
  for all  $S_i \subset S, \hat{S}_i \subset \hat{S}$ 
     $benefit = benefit + |\lambda_i - CAP(S_i)| - |\lambda_i - CAP(\hat{S}_i)|$ 
  return  $benefit$ ;

function update_S( $I, S, D, \Psi$ )
  for each  $AS_i \in I$ 
    find  $\bar{j}$  that satisfies  $Min_j(D_{i,j} + \Psi_j)$ 
    Assign  $AS_i$  to  $S_{\bar{j}}$ .
  return S;

```

Fig. 3. Load balancing algorithm

comparable, no matter how ASPP is applied. Therefore, we only need to keep track of comparable routes and examine an ASPP strategy's impact on them. Two paths are *comparable* if the preference between them can be changed by a padding action.

Figure 2 shows the algorithm to compute the sets of comparable routes across all N neighbors. The algorithm is a modification to the existing AS level path simulation [15]. We classify the links in the topology to be up link (customer-to-provider), down link (provider-to-customer) and flat link (peer-to-peer). A path which only follows UP links is called an uphill path. AS path "valley-free" policy permits AS paths in the form of *Customer-Provider* Peer-Peer? Provider-Customer**, where "*" represents zero or more occurrence of such type of AS edge and "?" represents at most one occurrence. Following this rule, the algorithm starts with the computation of the shortest uphill paths from the source node.

We use an $|I| \times |N|$ array D to keep all the best comparable paths between t 's neighbors in N and all other ASes in I . We first compute the shortest uphill paths from I to N in step 1. The function $update_dist_vec(k, D, PD)$ updates the distance from AS k to neighbors N according to AS relationship based policy. The uphill path contains only the customer-to-provider links. These links are always preferred over peer links and provider links [14]. Even if there was a much shorter path via peering link compared to an uphill path, the former would not be selected. If an AS can reach any of N , i.e., $\exists j, D_{kj} < \infty$, the search can stop because these paths will

```

1. function compute_backup( $I, D, u, v$ )# $I$ : set of ASes;  $u$ : primary neighbor
# $\Psi = \{\psi_1 \dots \psi_m\}$ : padding vector;  $v$ : back up neighbor
  for each  $AS_i \in I$ 
     $\delta_i = D_{iu} - D_{iv}$ , if  $D_{iu} > D_{iv}$ , otherwise  $\delta = 0$ 
    find  $\delta = \max_i(\delta_i)$ 
     $\psi_v = \psi_v + \delta + 1$ 

2. function compute_avoid_AS( $I, D, PD, \Psi, v$ )# $v$ : AS to bypass
  find all ASes ( $\{AS_i\}$ ) whose paths in  $PD$  traverse through  $v$ 
  for each  $AS_i$ 
     $\delta = \max_{u=1}^m (D_{iu} - D_{iv})$  if  $D_{iu} > D_{iv}$ , otherwise  $\delta = 0$ 
    find  $\delta = \max_i(\delta_i)$ 
     $\psi_v = \psi_v + \delta + 1$ 

```

Fig. 4. Padding vector computation for backup route and avoid AS

always be preferred. Otherwise, for the remaining ASes cannot reach N , we search the path via one peering link and update distance matrix D . Please note that if the path has already traversed one peering link from t to N^1 , then it cannot go through another peering link, according to the “valley-free” policy. Similarly, if a node still does not have a path via uplinks and flat link, we search all its providers recursively. The function $computeD(k, D, P)$ computes the distance in D from AS k to neighbors N through k 's providers or peers in P .

Please note that our algorithm considers the effectiveness of the AS-Path prepending technique together with the effect of general routing policies. The local routing policies are usually specified using Local preference attribute, which has a higher priority in the BGP path selection process than the AS-Path length [2]. Compared to existing work, our method is much closer to realistic scenarios.

The path matrix PD and distance matrix D now contain the best paths and their distances from all ASes to t 's neighbors N . Please note that PD only contains comparable paths, meaning that the prepending may affect the selection between them. In the previous example, only r_1 will be kept in the matrix PD . Next we use PD and D to compute $\Psi = \{\psi_i\}$. The assignment differs in details according to different objectives.

B. Performing load balancing

One common purpose of ASPP based traffic engineering is to balance the load across multiple providers and peers. Different providers may have different price and service level agreement. The operators in target AS t may have their own metrics in determining how much traffic shall traverse through which neighbor. We assume that the goal of traffic distribution among N is known $\Lambda = \{\lambda_i\}$. The operator adds ψ_i instances of padding at an advertised route to neighbor i until the AS-Path length is sufficient to reduce/increase the load from neighbor i to the expected value λ_i .

Figure 3 presents the algorithm that uses distance matrix D to assign the padding vector Ψ . We first partition all ASes(I) into m partitions $S = \{S_1 \dots S_m\}$, according their best route's first hop AS, t 's neighbor N . Some ASes only have a single route to t in matrix D , called single-path AS. For instance, AS_0 can only reach t through neighbor 2, then AS_0 can only be assigned to neighbor 2. After assignment, we remove AS_0

from I and deduce the traffic AS_0 generates from the capacity of neighbor 2. This process is shown in Step 1. In step 2, we iteratively increases the padding vectors. Each time, we choose to increase the ψ_j that maximize the benefit. After Ψ is updated, the assignment between AS and partition also needs adaptation in function $update_S$. The iteration stops until there is no benefit gain or it reaches the maximum iteration threshold.

C. Provisioning back-up path

For the purpose of provisioning backup routes, the degree of padding is usually large enough so that the backup route will not be adopted as best route unless if there is a failure in other primary routes. In this scenario, we compute the sufficient Ψ to ensure that a route is only used as a backup route of another. The first function in Figure 4, $compute_backup(I, D, u, v)$, computes the padding value to ensure that neighbor v will only be used as a back up for neighbor u . It simply searches the differences of $\delta = D_{iu} - D_{iv}$ for all cases v is preferred. The minimum padding value needed to make v less preferred in all ASes is one plus the maximum value of δ .

D. Avoid traversing particular ISP

Due to business reasons or political factors, the operators may want to avoid traffic traversing a particular AS. For instance, the countries known with information censorship issues are likely to be such examples. In the Facebook example, Facebook may want to avoid traffic traversing China Telecom. Using the computed path from the algorithm in Figure 2, we record all the comparable paths to t in PD . Searching through all paths in PD , we take a subset of them traversing v , the AS to bypass. Similarly, we compute the sufficient padding value from the maximum difference between path through v and all other paths.

In summary, we present our algorithm to discover and store comparable paths between all ASes to the target AS's neighbors. Then we use the stored path information to estimate the most suitable padding vector for three different TE goals.

V. CHARACTERIZATION RESULTS

We present the characterization of ASPP behaviors from BGP routing data below.

A. Analysis of prepending ASes' behavior

It is known that ASPP is commonly used for a non trivial portion of prefixes by most ASes [10]. We confirm this observation using more recent data in Mar. 2011. Figure 5 shows the fraction of prefixes whose best route contains ASPP, compared to the total number of prefixes. We calculate one fraction number for each monitor and present the CDF of many monitors. On average 13% routes in the default routing table have ASPP. Among different monitors, we do observe a significant difference. We suspect that edge ISPs are less likely to see prepending paths because the prepending paths are longer and thus are less likely to be selected as the best

ASN	Number of prefixes prepended	% of prefixes prepended	Name	Tier
6939	6991	37%	Hurricane Electric	2
1299	6161	3%	TeliaNet Global Network	1
3786	5811	78.5%	ERX-DACOMNET	2
4637	3493	1%	Reach Network	2
3257	3489	3%	Tiscali	2
9318	3014	35%	HANARO Telecom	2
6453	2902	1.5%	Teleglobe Inc	1
9121	2750	69%	TTnet	3
7545	2396	100%	TPG Internet	4
9002	1993	1.2%	RETN	3

TABLE IV
TOP 10 PREPENDING ASes IN ALL ROUTES IN THE DEFAULT-FREE ROUTING TABLE

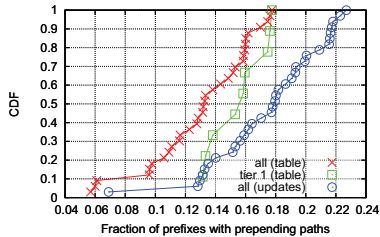


Fig. 5. Fraction of routes with prepending ASes

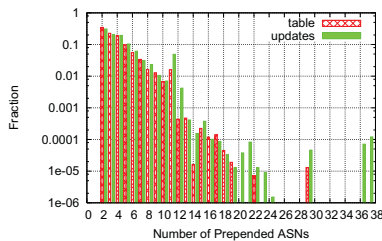


Fig. 6. Number of duplicate ASNs

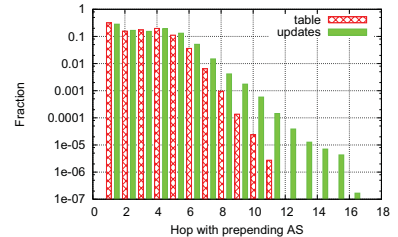


Fig. 7. Location of prepending AS in the path.

route. The top tier ISPs, on the other hand, may observe a larger fraction, since they can observe a diverse set of routes given their larger degrees. This conjecture is confirmed when we also plot the fractions for only Tier-1 ISPs in Figure 5. For similar reasons, in the update files, we also observe more routes with prepending ASes. In the unstable states, these routes are more likely to be visible in the route monitoring system.

Table IV shows the top 10 most frequent ASes which prepend their ASNs in routes of most prefixes. For each AS, we calculate the absolute number of prefixes where the AS prepend its own ASNs in the AS path. We then compare this absolute number with the total number of prefixes traversing this AS. The ASes are selected based on the absolute distinct number of prefixes. We observe two tier-1 ISPs actively prepending AS paths. The rest are mainly Tier-2 and Tier-3 ASes. The percentage of prefixes for Tier-1 ISPs is small, given a much larger number of routes traversing them. But the absolute value is non-negligible. Interestingly, AS7545 (TPG Internet) has a consistent policy to prepend for every prefix traversing it.

The operators configure the ASNs to be duplicated different number of times for different preference. For a less preferred route, it may repeat it many times to ensure it won't be chosen as the best route. We study how many repetition is common in most prepending AS paths. Figure 6 shows the number of duplicated ASNs in all the routes. Most of them are very small: 34% repeat twice and 22% repeat three times observed from routing table. 1% of them repeat larger than 10 times. The routes from update files have larger duplications. A related work has systematically measured other factors [16].

In the Facebook example, it is the origin AS prepending its

AS number multiple times. However, any AS along the paths can prepend its ASN for its own traffic engineering purposes. Figure 7 shows the first appearance of a chain of duplicated ASes. For instance, 31% of them are in the first hop, meaning that they are the origin ASes. Besides the origin AS, hop 2 to hop 5 have equal probability of prepending, on average 15%. There are no significant differences between updates and tables in most cases, except for a few instances of anomalously long paths.

For the same prefix, different ISPs may apply different routing policies. Therefore, route to the same prefix may be prepended by one AS but not by another. We study how consistent the prepending behavior is for each prefix in multiple ISPs. In Figure 8, we calculate the difference in the number of duplicated ASNs in routes from all monitors. For instance, if one monitor observes route $A B B B B$ and another observes $A B B$, then difference is 2. For each prefix, we take the maximum difference among all monitors and present the CDF of the differences. It shows that most prefixes are prepended consistently, i.e. with 0 difference.

Similarly, we show that for the same prefix, what fraction of observed routes have prepended ASNs in the path. Figure 9 shows the distribution of such fractions from a month of routing tables. The curve is evenly distributed, showing that the same prefix is not always announced with prepended AS paths.

B. Network impact

Figure 10 shows the monitors still choosing the route with prepended ASNs as the best route, in comparison with all monitors. Among 52 monitors, 70% of these ASPP are adopted by only 35% of the monitors. This is expected as

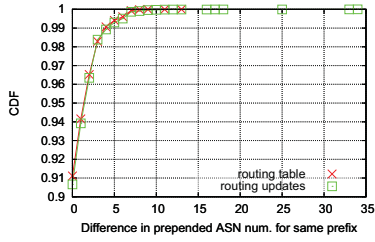


Fig. 8. Difference in number of prepended ASNs for the same prefix

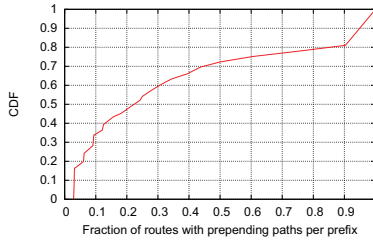


Fig. 9. Fraction of routes with prepending AS paths for the same prefix

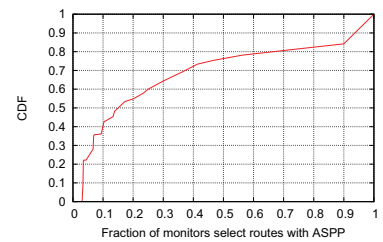


Fig. 10. Fraction of monitors adopting routes with ASPP.

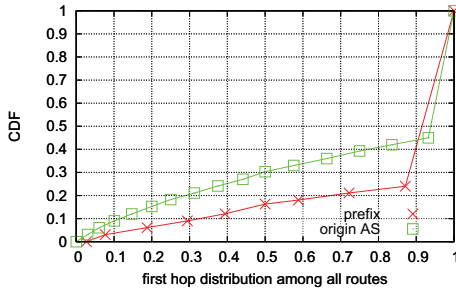


Fig. 11. First hop distribution among neighbors

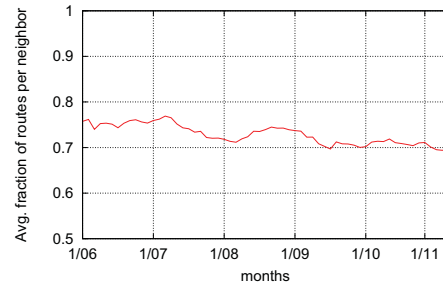


Fig. 13. Trends of route distribution amongst neighbors over time

the prepending AS paths are likely to be longer than other paths, thus less preferable. However, there are 20% of the routes are still adopted by most of the monitors. With further investigation, we found that it's the origin AS that prepends its ASNs multiple times consistently for announcements to all its neighbors. Therefore, even the prepending increases the AS path length, it is an equal inflation for all monitors.

ASPP can be used to balance the traffic among all neighbors or to provision a backup route. We categorize routes from all monitors for the same prefix at any time according to their first hop ASes. We then calculate the fraction of routes associated with each first hop. The distribution of such fraction numbers are shown in Figure 11. Surprisingly, we see that 90% cases there is only one dominant route (the fraction is 1) for each prefix. But examining from all prefixes of same origin AS, we observe a much smaller fraction. It suggests that operators commonly use ASPP to make best routes go through different neighbors for different prefixes.

C. Dynamic characterization

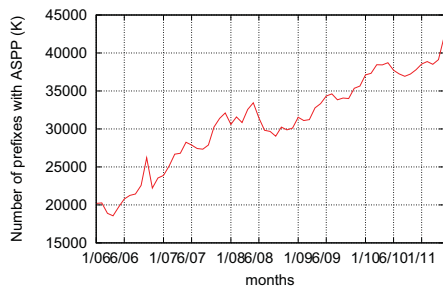


Fig. 12. Trends of routes with ASPP in the default-free routing table

We next demonstrate how the prepending AS path involves over time. Figure 12 shows the number of prefixes with any prepending AS paths from January 2006 to March 2011 from RouteView. It presents significant increases, suggesting that ASPP as a traffic engineering method has been more commonly used. We next depict the trends of the impact of ASPP. Figure 13 shows the average fraction per month over five years. The trend is very stable, *i.e.*, 70%-80%, suggesting that each prefix often has a primary carrier amongst all neighbors.

VI. VALIDATION

In the following, we first validate the accuracy of our prediction on route selection. Then we evaluate the algorithms' effectiveness from three aspects, *i.e.*, load balancing, backup route provisioning and bypassing a particular AS. The topology is extracted from multiple BGP routing tables and the topology reported by CAIDA [17].

A. Validation of route prediction

Since we cannot directly modify the prepending behavior on the Internet, we evaluate the simulation accuracy by comparing the predicted routes with the adopted routes on the Internet. One of the key steps in our proposal is the prediction of routes seen and chosen by each AS in Figure 2. We validate its accuracy using observed best routes from RouteView. From one month of RouteView table and update files, we extract all the prefixes which is announced to multiple neighbors by the origin AS padding its ASN different times. We plug the observed padding numbers for each neighbor into the algorithm in Figure 2 to compute the best route observed on each AS. For each prefix and padding vector, we compute the fraction of ASes where the predicted route matches with the

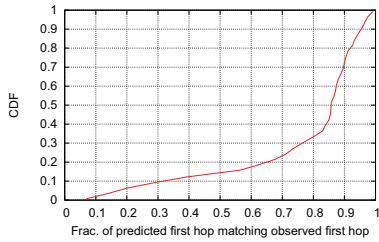


Fig. 14. Prediction accuracy compared to observed routes

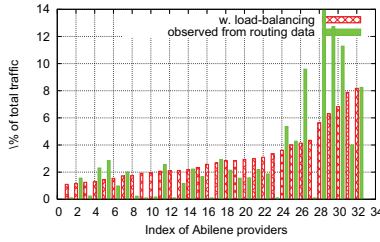


Fig. 15. Traffic distribution among Abilene's providers

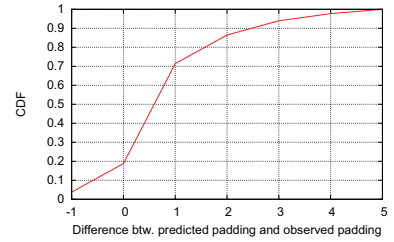


Fig. 16. Number of padding ASNs for backup route provisioning

observed one. Figure 14 shows the CDF of the overlapping fraction. In 70% of the cases, our prediction is above 80%. Please note that the accuracy is in general higher than the accuracy of AS path prediction [18]. This is because we only require the first hop of the predicted route matches with the observed first hop, i.e. which neighbor an AS chooses to go through.

B. Evaluation of load-balancing

To illustrate the effectiveness of our algorithms in various goals of traffic engineering, we apply it on Abilene network whose AS-level 32 providers. Abilene [19] is the U.S. Internet backbone for educational networks. Abilene maintains peering relationship with other educational networks and multi-homes to many major ISPs. We obtain netflow records for one week from Abilene all 11 core routers. From the netflow data, we estimate the traffic matrix from all other ASes to Abilene. From netflow, we first compute the amount of traffic originated from each AS. Given 32 neighbors, we set the capacity constraints to be $\frac{1}{32}$ of the total traffic. Using Algorithm in Figure 3, we compute the best padding vector Ψ for Abilene to achieve the balanced load among all its 32 neighbors. Figure 15 shows the relative traffic going through each neighbor, in comparison with the observed distribution from netflow. With our algorithm, we can achieve much better balance compared to existing distribution.

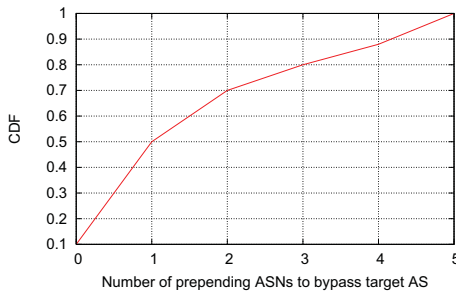


Fig. 17. Number of padding ASNs for target AS bypassing

C. Evaluation of backup route provisioning

As shown in Section V, today some ASes already use ASPP to provision backup path. However, the selection of number of padding ASNs is rather random. We take this subset of ASes and recompute the minimum amount of padding ASNs

needed to guarantee the neighbor will be less preferred than other paths. Figure 16 shows the PDF of minimum number of padding instances from the computation for each instance, in comparison of the prepending number being used today. It is shown that in most cases, we reduce the padding number by 1 and 2. Although it does not affect the traffic distribution, it shows that random prepending may result in unnecessarily long paths.

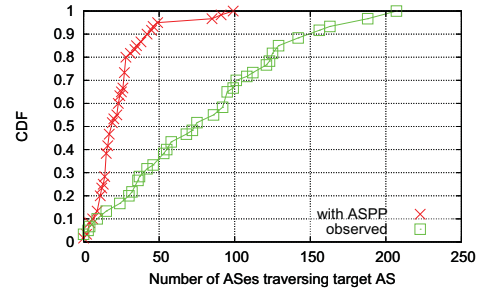


Fig. 18. ASes traversing target AS bypassing w/o ASPP

D. Evaluation of bypassing an AS

Since the Great Firewall information censorship is of large concerns to many content providers, we use China Telecom (AS4134) as an example of the AS to bypass. For instance, a popular web site such as Google or Facebook may want to avoid traffic destined to it going through China Telecom. We take the ASes hosting the list of top 50 websites [20], and study how many AS should be prepended to minimize the routes traversing through AS4134. The number of prepending instances are shown in Figure 17. For most origin ASes, prepending 2 ASes can be sufficient. We further present how many ASes still traverse the target AS, in comparison to observed paths from routing data directly. Figure 18 shows that on average we can reduce the number of ASes traversing target AS by 42%.

VII. RELATED WORK

The Border Gateway Protocol (BGP) [21] is the inter-domain routing protocol that Autonomous Systems (ASes) use to exchange information about how to reach destination address blocks (or *prefixes*). There exist several public route monitoring systems, such as Route Views [12] and RIPE [13], to help understand and monitor the Internet routing system.

Various research studies have been conducted relying on the BGP data, including network topology discovery [22], AS relationship inference [23], [24], [25], [26], [14], AS-level path prediction [18], [27], BGP root cause analysis [28], and several routing anomaly detection schemes. Local BGP changes in a single AS may propagate globally and result in routing changes in all the ASes on the Internet. Feamster *et al.* [29] applied static analysis to find faults in BGP configurations. Previous measurement studies [30], [31], [32] have already shown that routing changes can cause transient disruption to the data plane.

Our work is more close to the previous work on AS path prepending. Chang *et al.* [8] proposed a systematic procedure to predict the changes in traffic distribution for a given ASPP configuration. They proposed a system to automatically determine the number of ASes to prepend for a multi-homing AS. The goal is to prevent routing instability. Our goal is general traffic engineering tasks. Wang *et al.* [10] studied the instability problems of ASPP approach and provided guidelines for convergence. In [7], the authors used an optimization based approach to determine the optimal padding at each ingress link so that the constraints are met.

Finally, the characterization of AS path prepending behavior is within the category of routing policy inference. There is much work [33], [26], [23], [14], [24] on inferring AS relationships from BGP AS paths. Our work builds on top of their results as well as the algorithms for predicting AS level paths [18], [15], [34]. Our work is complementary to the above work as we focus on a different aspect of the routing policies, the AS path prepending.

VIII. CONCLUSIONS

AS Path prepending is a common approach for inter-domain traffic engineering. It relies on manipulating the AS path length by purposely inserting its own ASN multiple times. In this work, we present a comprehensive study on the characterization of Internet routing AS path prepending. Our results illustrate the popularity and effectiveness of ASPP as a TE mechanism. We further propose an algorithm for computing the optimal padding vectors for the operators. Our method considers the impact of AS relationship based local policies on ASPP's effectiveness. The algorithm can be used for three objectives, *i.e.*, traffic load balancing, backup route provisioning, and specific AS bypassing. Our validation shows that the prediction can achieve above 80% accuracy for 70% cases. We further demonstrate the effectiveness using realistic Abilene traffic and topology. In our future work, we plan to develop techniques to detect any misconfiguration or malicious attacks by exploring the ASPP phenomenons.

REFERENCES

[1] N. Feamster, J. Borkenhagen, and J. Rexford, "Guidelines for interdomain traffic engineering," *SIGCOMM Comput. Commun. Rev.*, vol. 33, pp. 19–30, October 2003.
 [2] Cisco, "Bgp best path selection algorithm." <http://www.cisco.com/warp/public/459/25.shtml>.

[3] M. Howarth, M. Boucadair, P. Flegkas, N. Wang, G. Pavlou, P. Morand, T. Coadic, D. Griffin, H. Asgari, and P. Georgatsos, "End-to-end quality of service provisioning through inter-provider traffic engineering," *Computer Communications*, vol. 29, pp. 683–702, March 2006.
 [4] D. Goldenberg, L. Qiu, H. Xie, Y. Yang, and Y. Zhang, "Optimizing cost and performance for multihoming," in *Proc. ACM SIGCOMM*, 2004.
 [5] B. Quoitin, S. Uhlig, C. Pelsser, L. Swinner, and O. Bonaventure, "Inter-domain traffic engineering with bgp," *IEEE Communications Magazine*, vol. 41, p. 122 C 128, May 2003.
 [6] Y. Yang, H. Xie, H. Wang, A. Siberschatz, A. Krishnamurthy, Y. Liu, and L. Li, "On route selection for interdomain traffic engineering," *IEEE Network Magazine*, vol. 19, pp. 20–27, November 2005.
 [7] R. Gao, C. Dovrolis, and E. Zegura, "Interdomain ingress traffic engineering through optimized as-path prepending," in *Proc. of IFIP Networking conference*, 2005.
 [8] R. Chang and M. Lo, "Inbound traffic engineering for multihomed as's using as path prepending," *IEEE Network*, vol. 41, pp. 18–25, April 2005.
 [9] G. Battista, M. Patrignani, M. Pizzonia, and M. Rimondini, "Towards optimal prepending for incoming traffic engineering," in *Proc. of IPS-MoMe*, 2005.
 [10] J. H. Wang, D. M. Chiu, J. C. S. Lui, and R. K. C. Chang, "Inter-as inbound traffic engineering via aspp," *IEEE Transaction on Network and Service Management*, vol. 4, June 2007.
 [11] "Hey ATT customers: Your Facebook data went to China and S. Korea this morning." www.blyon.com/hey-att-customers-your-facebook-data-went-to-china-and-korea-this-morning.
 [12] "University of Oregon Route Views Archive Project." www.routeviews.org.
 [13] "Ripe NCC." <http://www.ripe.net/ripenncc/pub-services/np/tris/>.
 [14] L. Gao, "On Inferring Autonomous System Relationships in the Internet," in *Proc. IEEE Global Internet Symposium*, 2000.
 [15] J. Wu, Y. Zhang, Z. M. Mao, and K. G. Shin, "Internet routing resilience to failures: analysis and implications," in *Proceedings of the 2007 ACM CoNEXT conference*, CoNEXT '07, pp. 25:1–25:12, 2007.
 [16] Y. Zhang and M. Tatipamula, "A comprehensive long-term evaluation on bgp performance," in *Proceedings of IEEE International Conference on Communications (ICC)*, 2011.
 [17] "NetGeo - The Internet Geographic Database." <http://www.caida.org/tools/utilities/netgeo/index.xml>.
 [18] Z. M. Mao, L. Qiu, J. Wang, and Y. Zhang, "On AS-Level Path Inference," in *Proc. ACM SIGMETRICS*, 2005.
 [19] "The internet2 network." www.internet2.edu/network/.
 [20] "Alexa web wearch top 500." www.alexa.com/topsites.
 [21] Y. Rekhter and T. Li, "A Border Gateway Protocol." RFC 1771, March 1995.
 [22] Y. He, G. Siganos, M. Faloutsos, and S. V. Krishnamurthy, "A systematic framework for unearthing the missing links: Measurements and Impact," in *Proc. of NSDI*, 2007.
 [23] X. Dimitropoulos and G. Riley, "Modeling Autonomous System Relationships," in *Proc. of PADS*, 2006.
 [24] X. Dimitropoulos, D. Krioukov, M. Fomenkov, B. Huffaker, Y. Hyun, kc claffy, and G. Riley, "AS Relationships: Inference and Validation," *ACM Computer Communication Review*, vol. 37, no. 1, 2007.
 [25] G. Battista, M. Patrignani, and M. Pizzonia, "Computing the Types of the Relationships Between Autonomous Systems," in *Proc. IEEE INFOCOM*, March 2003.
 [26] L. Subramanian, S. Agarwal, J. Rexford, and R. H. Katz, "Characterizing the Internet hierarchy from multiple vantage points," in *Proc. IEEE INFOCOM*, 2002.
 [27] W. Muhlbauer, A. Feldmann, O. Maennel, M. Roughan, and S. Uhlig, "Building an AS-Topology Model," in *Proc. of ACM SIGCOMM*, 2006.
 [28] A. Feldmann, O. Maennel, Z. M. Mao, A. Berger, and B. Maggs, "Locating Internet Routing Instabilities," in *Proc. ACM SIGCOMM*, 2004.
 [29] N. Feamster and H. Balakrishnan, "Detecting BGP Configuration Faults with Static Analysis," in *Proc. 2nd Symposium on Networked Systems Design and Implementation (NSDI)*, May 2005.
 [30] C. Labovitz, A. Ahuja, A. Bose, and F. Jahanian, "Delayed Internet Routing Convergence," in *Proc. ACM SIGCOMM*, 2000.
 [31] F. Wang, Z. M. Mao, J. Wang, L. Gao, and R. Bush, "A Measurement Study on the Impact of Routing Events on End-to-End Internet Path Performance," in *Proc. ACM SIGCOMM*, 2006.
 [32] Z. M. Mao, R. Bush, T. G. Griffin, and M. Roughan, "BGP beacons," in *Proc. ACM SIGCOMM Internet Measurement Conference*, 2003.
 [33] G. Battista, M. Patrignani, and M. Pizzonia, "Computing the Types of the Relationships Between Autonomous Systems," in *Proc. IEEE INFOCOM*, March 2003.
 [34] W. Muhlbauer, A. Feldmann, O. Maennel, M. Roughan, and S. Uhlig, "Building an AS-topology model that captures route diversity," in *Proc. ACM SIGCOMM*, 2006.