vDC: Virtual Data Center Powered with AS Alliance for Enabling Cost-Effective Business Continuity and Coverage

Yuichiro Hei
KDDI R&D Laboratories, Inc.

Akihiro Nakao
The University of Tokyo

Tomohiko Ogishi
KDDI R&D Laboratories, Inc.

Toru Hasegawa
KDDI R&D Laboratories, Inc.

Shu Yamamoto
NICT

Abstract

In the cloud computing era, cloud providers must design data centers that satisfy the requirements such as business continuity, coverage and performance, and cost-effectiveness for offering application providers the competitive hosting services. However, it is hard for even elephant cloud providers to satisfy these requirements all together because of the cost problem. In this paper, we propose the concept of *virtual* data center (*v*DC) of multiple geographically distributed data centers over the Internet to extend the coverage of hosting services in a cost-effective manner and apply the concept of *AS alliance* to ensure resilient connectivity of vDC to achieve high business continuity. We also introduce the detail design of AS alliance tailored for the vDC concept and conduct a feasibility study for making vDC connectivity robust.

1 Introduction

Recently, more and more applications and data are being served from carefully designed large-scale data centers, a.k.a., cloud computing platforms, over the Internet, since we tend to care how fast and reliable our access is to the services rather than *where* and *how* they are hosted [10, 12]. In this cloud computing era, application providers pay cloud providers that operate data centers for hosting their applications to serve users. Accordingly, it has become clear that cloud providers must design data centers that satisfy the following three crucial requirements for enabling competitive hosting services for application providers: (1) *business continuity*—continuously and reliably host mission-critical network services even when catastrophic hardware failures and natural disasters occur, (2) *coverage and performance*—widely cover geographically diverse users, thus, provide the users fast and reliable access to the hosted services (3) *cost-effectiveness*—minimize cost for hosting services, which eventually leads to providing users with inexpensive access to them.

There exist a multitude of cloud providers ranging from small regional ones to large planetary-scale ones, or in another perspective, from centrally operated ones to distributed ones inter-connected via resilient network connections. However, no matter which design to choose to realize business continuity and coverage, the problem eventually boils down to the cost for achieving them. We observe that a centrally-managed, large data center is generally expensive in terms of processing and network resources to realize business continuity. For example, a new middle-size data center with about 54,000 servers and 13.5MW (250 Watts/server $\times$ 54K servers) costs over $200M [5]. In addition, provisioning network bandwidth through peering with tier-1 networks is necessary due to data traffic implosion, but is also costly. For example, it is reported that Google is losing more than $1M per day for bandwidth due to a huge volume of YouTube traffic concentrated on their data centers [9]. Similarly, it is hard for a single cloud provider to shoot the two birds—business continuity and footprint—at the same time, e.g., extending business coverage through constructing data centers in different continents while ensuring data synchronization and replication among them requires investment on fast and resilient network connectivity. For example, recently, Google has jointly filed the contract of laying a trans-oceanic submarine fiber for about $300M [7]. While even elephant cloud providers such as Google and Amazon are facing the cost problem, it is almost prohibiting for small regional ones to play the same game for business continuity and coverage in a cost-effective manner.

Our goal is to conduct a cost-effective way for small regional data centers to scale out into a global data center to satisfy the requirements for data centers. In this paper, we posit that constructing a *virtual* data center (*v*DC) of multiple geographically distributed data centers operated by different organizations over the Internet and *ensuring robust connectivity among them* is the key to cost effectively scaling out to the global business cov-

erage and achieving business continuity. In a nutshell, our idea is to separate *cloud service providers* and *data center providers* to allow the former to build a vDC on top of the resources purchased from the latter and to connect them through multiple disjoint paths over the Internet. Exploiting the existing data center infrastructures without any dedicated links between them not only saves the cost for constructing a new infrastructure but also enables dynamically control the extent of business coverage and reconfiguring connectivity for data synchronization and replication. Our contributions in this paper are two-fold: first, we propose the concept of vDC to extend the footprint of hosting services in a cost-efficient manner. Second, in putting into practice the concept of vDC spanning across multiple ASes, we apply the concept of *AS alliance* [8] that discovers multiple disjoint paths among the selected ASes to achieve high business continuity over the Internet in an inexpensive way despite the limitation of BGP [16, 15]. We introduce the detail design of AS alliance tailored for the vDC concept and conduct a feasibility study for making vDC connectivity robust. Our vDC design powered by the AS alliance connectivity mechanism slightly extends BGP and uses IP tunnels, thus, is practical enough to be deployed today in the current Internet.

The rest of the paper is organized as follows. Section 2 reports related work. Section 3 introduces the concept of our vDC architecture over AS alliance and Section 4 describes its design. Section 5 demonstrates our prototype implementation of vDC over AS alliance and Section 6 briefly concludes.

2 Related Work

An approach to separating infrastructure providers and service providers has been presented in Mobile Virtual Network Operators (MVNO) and Cabo [6]. The purpose of this virtual infrastructure model is to optimize the resource allocation, especially reducing cost by efficiently sharing physical infrastructures and to bring a new business model into the market. We show that the same virtual infrastructure model is also applicable to the cloud hosting service business, i.e., cloud service providers can purchase resources and infrastructures such as housing space, processing power, network connectivity from data center providers.

There exist a few studies that discuss similar concepts of alliance among ASes [13, 18]. Although, in these studies, an alliance is formed among neighboring ASes, our AS alliance is formed among arbitrary (e.g., remote) ASes. R-BGP [14] is an approach to ensures that multiple ASes stay connected as long as the Internet is connected. R-BGP provides pre-computed failover paths to get around a failed link rapidly, but does not provide multiple paths. More importantly, R-BGP improves only unilateral connectivity from any AS to a target AS *X* and requires *externality*—cooperation from several other parties that not directly benefit from the system. For instance, only if AS *B* knows alternative paths to AS *X* and a set of ASes located along the primary path from AS *B* to AS *X cooperatively* install R-BGP for the target AS *X*, R-BGP can quickly failover a link failure on the primary path to *X*. Therefore, several ASes must share incentives to deploy R-BGP for the sake of improving resilience to AS *X*. In contrast, our alliance approach brings benefit—achieving resilience in mutual communications—to *all and only* the alliance members, thus minimizing externality and avoiding incentive issues to hinder its deployment.

SBone [2] aims to secure BGP. In SBone, an AS overlay network is formed among a small group of ASes by a mesh of virtual links. Also, just running eBGP sessions overlaid on top of an AS virtual network may appear similar to AS alliance. As far as we know, however, we first propose to apply an AS overlay to inter-connect data centers over the Internet.

3 Architecture of vDC over AS Alliance

3.1 Overview

In our proposal, we assume a cloud service provider purchases resources from multiple (several, most likely three) data centers and consolidates them into a virtual data center over the Internet. The separation of data center providers and cloud service providers thus extends the business coverage of a single cloud service provider without incurring the cost of introducing new processing and network infrastructures. However, the question is how to achieve business continuity in a cost-effective way over the Internet, where AS alliance comes into play to the rescue to resolving the issue.

Figure 1 depicts the overview of the architecture of vDC over AS alliance. It shows the Internet at the bot-

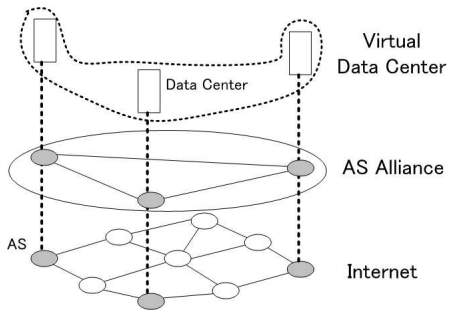

Figure 1: Virtual data center over AS alliance.

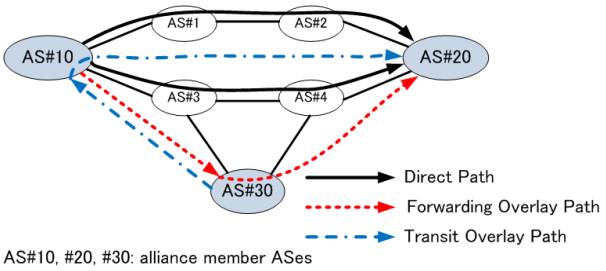

Figure 2: The three types of path from the viewpoint of AS#10 in the alliance.

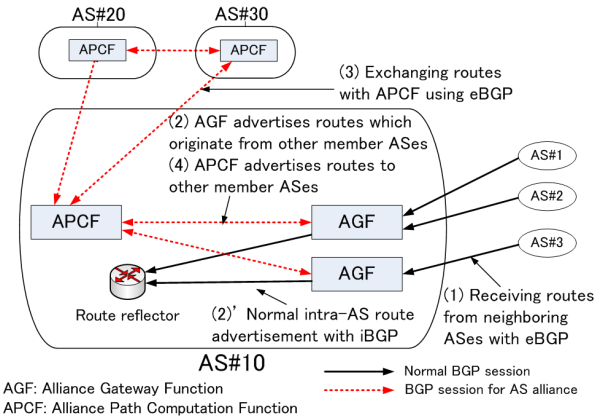

Figure 3: Architecture inside an alliance member AS.

tom, an AS alliance on top of the Internet, and a vDC over the AS alliance. Data centers forming a vDC operate in different ASes and exchange route information via BGP. AS alliance is formed by a given set of ASes in the Internet. It provides robust communication among the alliance member ASes over the Internet. Therefore, forming an AS alliance among ASes of data centers consisting of a vDC makes the inter-communication among them resilient.

In order to realize the robust communication among the alliance member ASes, an AS alliance provides multiple paths among the member ASes. Each member AS constructs multiple paths by sharing BGP routes[1] with one another and a member AS provides the other members with a transit between them. Moreover, to prevent the multiple paths from becoming vulnerable simultaneously due to a single point of failure, e.g., a link failure, a member AS computes the multiple paths to be as disjoint as possible from the BGP routes. We assume that a member uses one of the multiple paths as the primary one and others as backup, or it may use the multiple paths simultaneously. We also assume that AS alliance members are edge ASes, i.e., they do not transit traffic between other ASes in the normal routing. Note that throughout this paper, we describe a three-AS alliance as an example of an AS alliance, since a three-AS alliance is the simplest possible form of alliance and also can be an essential building block of further extension [8].

3.2 Paths among the AS alliance members

We define the three types of paths in an AS alliance. Figure 2 illustrates these three types of paths *from the viewpoint of AS#10* in the alliance where the members are AS#10, #20 and #30.

The first path is a direct path, that is, a normal BGP path. The second path is a forwarding overlay path via AS#30. Even if all direct paths fail, AS#10 can continue to communicate with AS#20 over this overlay path with help from AS#30. The third path is a transit overlay path that AS#10 provides, which is a portion of the forwarding overlay path from AS#30 to AS#20 via AS#10.

The reason that a transit overlay path is needed is as follows. To forward packets from AS#10 to AS#20 along the forwarding overlay path, the origin router of AS#10 needs to force packets to AS#20 to traverse AS#30. The simplest way to realize this is tunneling; a packet from AS#10 to AS#20 is encapsulated by adding a header whose source is an address of AS#10 and destination is an address of AS#30, and it is decapsulated in AS#30 and forwarded to AS#20. However, if upstream ASes of AS#30 implement strict reverse path forwarding [3], a packet originating from AS#10 may be dropped in the upstream ASes because the source address of the packet is not in AS#30. In this case, AS#30 cannot use direct paths to forward this packet to AS#20. To bypass this filtering, AS#30 encapsulates this packet again by adding a header whose source is an address of AS#30 and destination is an address of AS#20. To distinguish a direct path and this tunneling path, we define this tunnel as a transit overlay path.

3.3 Inside a member AS

Figure 3 illustrates the system design at each alliance member. Each member installs two types of software components: an alliance path computation function (APCF) and (multiple) alliance gateway functions (AGFs). In Fig. 3, a solid arrow means a normal BGP session, and a dashed arrow means a BGP session for establishing an AS alliance.

An AGF receives routes from neighboring ASes with eBGP (Fig. 3(1)) and advertises only the routes that orig-

[1]In this paper, the term "route" generally means the information that will be advertised in a BGP update message, and the term "path" is used as an instance of the route.

inate from the other members to the APCF in the same AS (Fig. 3(2)). The reason to apply this filtering is that an APCF needs to obtain only AS paths between arbitrary pair of the members. An AGF also has normal BGP router functions, i.e., it advertises routes to normal BGP peers in its belonging AS with iBGP (Fig. 3(2)').

The APCF exchanges routes that are received from the AGFs of its own AS with the APCFs of the other members (Fig. 3(3)). By this route exchange, the APCF knows the AS paths between arbitrary pair of its alliance members. This exchange can be done with establishing full-mesh BGP sessions among the APCFs of the members. Alternatively, the APCF establishes a BGP session with only the aggregating APCF, which works as a route reflector. The APCF then computes disjoint paths among alliance members, and advertises BGP update messages to the AGFs (Fig. 3(4)). When an AGF receives BGP update messages from the APCF, it constructs routing tables used for packet forwarding to the alliance members.

4 Design

This section elaborates the design of an AS alliance, especially how an AS alliance discovers and utilizes multiple routes. We then describe how an APCF computes the paths and advertises them to an AGF and how the AGF constructs its routing tables.

4.1 Multiple routes

An AGF may receive multiple routes to each prefix of the members from neighbors as shown in Fig. 4. An AGF advertises each update to the APCF without selecting the best route to each prefix because we collect as many AS paths as possible to improve the probability of finding disjoint paths among the members.

Since the APCF must distinguish these multiple updates destined to the same prefix received from an AGF, the AGF annotates BGP updates with path identifiers as in Fig. 4. In general, a path identifier is prepended in network layer reachability information (NLRI) [17]. We make a semantically different use of path identifiers, i.e., to differentiate updates from multiple neighbors.

4.2 Path computation and update

An APCF computes disjoint paths among alliance members using routes received from the AGFs in the same AS and the APCFs in the other members [8]. The APCF recomputes disjoint paths among alliance members if it detects a change in AS path information, e.g., a new AS path appearing or the existing AS path withdrawing. After computing disjoint paths, an APCF composes BGP update messages to advertise the routes to the AGFs.

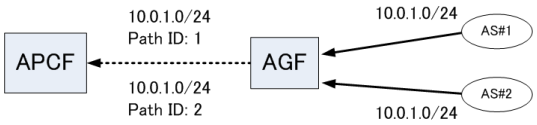

Figure 4: Advertising multiple routes for the same prefix with path identifier.

To elaborate how an APCF advertises BGP update messages to the AGFs, we revisit the same example shown in Fig. 2. Let us assume that the APCF in AS#10 knows multiple paths to a prefix in AS#20, i.e., a direct path and a forwarding overlay path via AS#30. The APCF must notify these paths to the AGFs distinctly. Furthermore, it must notify a transit overlay path to the AGFs so that AS#30 is able to forward packets to AS#20 via AS#10. We use the community attribute [4] to distinguish these paths in update messages.

4.3 Routing in the AGF

The routing among alliance members should be separated from the normal routing. We define three routing tables in the AGF. The first one is for the normal routing, the second one is for forwarding packets from its own AS to the alliance members (routing table for its own AS), and the third one is for transiting packets between other members (routing table for the other ASes).

When an AGF receives a BGP update message from the APCF in the same AS, it checks the community attribute to select the routing table to update. If an update message includes no community value or the one assigned for a forwarding overlay path, an AGF updates the routing table for its own AS. If it includes the value assigned for a transit overlay paths, an AGF updates the routing table for other ASes. When an AGF forwards a packet, it checks the source address of the packet to decide the routing table to refer.

An AGF implements tunnel interfaces to the other members to set up the overlay paths. Figure 5 illustrates the tunnel paths to the other members in the routing tables of the AGF in AS#10. For example, the AGF in AS#10 registers the forwarding overlay path toward AS#20 to the routing table of its own AS and this path is associated with the tunnel interface to AS#30. On the other hand, it registers the transit overlay path toward AS#20 to the routing table for the other ASes and this path is associated with the tunnel interface to AS#20. As shown in Fig. 5, a tunnel can be shared between each of these overlay paths.

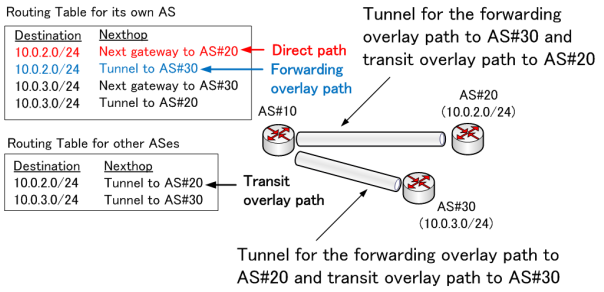

Figure 5: Routing tables and tunnels to the other members in an AGF.

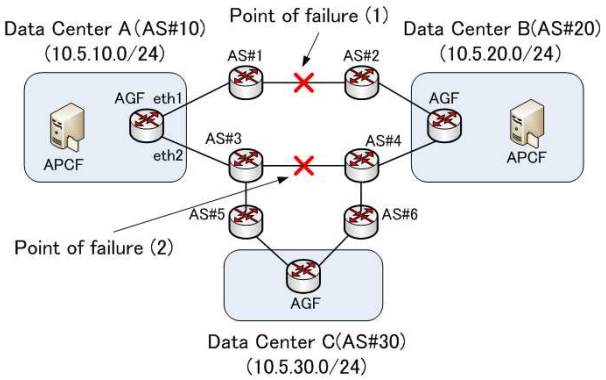

Figure 6: Evaluation topology.

5 Prototyping and Evaluation

We have implemented a prototype of AS alliance on Linux boxes and conducted a preliminary evaluation. Our implementation requires only a slight modification to the existing software. Since we can set up multiple IP routing tables on Linux, we can simply set up routing tables in the AGF as described in Section 4.3 and add a feature to Quagga BGP routing daemon [1] for advertising multiple routes with path identifiers as described in Section 4.1.

Figure 6 shows the topology used for our evaluation. Data center (DC) A, B, and C are edge ASes that form an AS alliance. A tunnel from an AGF to the other member DC is terminated at a terminator (not shown in Fig. 6). In this environment, we evaluate how data replication with *rsync* command between DC-A and B is performed. Simultaneous link failures are caused artificially at the links (1) and (2) in Fig. 6 during the data replication. In section 5.1, we show how the routing table in the AGF of DC-A changes as a result of the link failures. In section 5.2, we show the results of *rsync* with and without AS alliance.

5.1 Routing table

This subsection describes how routing tables change to mask link failures between member ASes. Figure 7(a) shows an example of a part of routing table for its own AS in the AGF of DC-A before the link failures are occurred. The figure shows the multiple paths from DC-A to DC-B and DC-C: one is a direct path and the other is a forwarding overlay path (note that the device "tun10to20" is a tunnel interface from AS#10 to AS#20). The metric of a direct path is smaller than that of a forwarding overlay path, so normally the direct path is used as a primary path.

Figure 7(b) shows the routing table after the link failures are occurred. Although all the direct paths from DC-A to DC-B disappear, DC-A can continue to communicate with DC-B because a forwarding overlay path to DC-B via DC-C is alive.

```
# ip route show table 10
10.5.30.0/24 via 172.31.2.3 dev eth1 proto zebra metric 1
10.5.30.0/24 dev tun10to20 proto zebra metric 10
10.5.20.0/24 proto zebra metric 1
        nexthop via 172.31.2.3 dev eth1 weight 1
        nexthop via 172.31.3.3 dev eth2 weight 1
10.5.20.0/24 dev tun10to30 proto zebra metric 10
```

(a) Normal state

```
# ip route show table 10
10.5.30.0/24 via 172.31.2.3 dev eth1 proto zebra metric 1
10.5.20.0/24 dev tun10to30 proto zebra metric 10
```

(b) Failure state

Figure 7: Routing table in the AGF of DC-A.

5.2 Data replication

Figure 8 shows parts of the results of *rsync* command on the host in DC-A when the direct paths between DC-A and B disappear during data replication from DC-A to DC-B . Figure 8(a) shows how *rsync* is executed in the case that the AS alliance is formed among DC-A, B and C, and Figure 8(b) shows in the case that an AS alliance is not formed among these DCs.

With AS alliance, the data replication has successfully finished even if the direct paths disappear between DC-A and B. In this case, the communication between DC-A and B can continue via a forwarding overlay path. On the other hand, without AS alliance, the data replication has

```
# rsync -avh –progress /root/testfile 10.5.20.1:/root/
building file list ...
1 file to consider
testfile
    1.02G 100% 5.85MB/s 0:02:45 (xfer#1, to-check=0/1)

sent 1.02G bytes received 42 bytes 4.65M bytes/sec
total size is 1.02G speedup is 1.00
```

(a) rsync finishes successfully if AS alliance is formed among DC-A, B, and C.

```
# rsync -avh –progress /root/testfile 10.5.20.1:/root/
building file list ...
1 file to consider
testfile
Read from remote host 10.5.20.1: No route to host
rsync: writefd_unbuffered failed to write 4 bytes ...
rsync: connection unexpectedly closed ...
rsync error: unexplained error (code 255) ...
```

(b) rsync fails if AS alliance is *not* formed among DC-A, B and C.

Figure 8: Results of rsync when the direct paths disappear during the data replication between DC-A and B.

failed because there is no route between DC-A and B. These results show that AS alliance can provide the robust communication among data centers forming a vDC.

In AS alliance, the downtime of a single path depends on the time of failure detection and route recomputation. However, this can be mitigated if a faster failure detection technique such as a bidirectional forwarding detection [11] is implemented in BGP routers. Furthermore, the application can avoid the influence of a failure of the single path if multi-path application protocols are implemented.

6 Conclusions

In this paper, we propose a *virtual* data center (*v*DC) consisting of multiple geographically distributed data centers operated by different organizations over the Internet by applying the concept of AS alliance to *ensure robust connectivity among individual data centers* to cost effectively realize global business continuity and coverage. We present the practical design of an architecture of vDC over AS alliance and implement its prototype to conduct a feasibility study for making vDC connectivity robust. Our vDC design powered by the AS alliance connectivity mechanism only slightly extends BGP and IP tunnels and is practical enough to be deployed in the current Internet. Our preliminary study shows that vDC with AS alliance can provide the robust communication among data centers forming a vDC.

There are quite a few interesting topics for our future work. Among them, first, we plan to work on the control interface for a cloud service provider to dynamically reconfigure a vDC. Second, we will analyze the stability of a vDC over AS alliance in case of instability in the Internet, e.g., route flapping, etc. Finally, we plan to deploy a vDC in the real Internet and evaluate its effectiveness.

Acknowledgment

This work was partly supported by Ministry of Internal Affairs and Communications of the Japanese Government.

References

[1] Quagga software routing suite. http://www.quagga.net.

[2] AVRAMOPOULOS, I., SUCHARA, M., AND REXFORD, J. How small groups can secure interdomain routing. http://www.cs.prinston.edu/research/techreps/TR-808-07.

[3] BAKER, F., AND SAVOLA, P. Ingress filtering for multihomed networks. RFC 3704, March 2004.

[4] CHANDRA, R., TRAINA, P., AND LI, T. BGP communities attribute. RFC 1997, August 1996.

[5] CHURCH, K., GREENBERG, A., AND HAMILTON, J. On delivering embarrassingly distributed cloud services. In *Proc. of Hotnets* (October 2008).

[6] FEAMSTER, N., GAO, L., AND REXFORD, J. How to Lease the Internet in Your Spare Time. *ACM SIGCOMM Computer Communications Review* (January 2007), 61–64.

[7] GOOGLE PRESS. Global Consortium to Construct New Cable System Linking US and Japan to Meet Increasing Bandwidth Demands. http://www.google.com/intl/en/press/pressrel/20080225_newcable system.html.

[8] HEI, Y., NAKAO, A., HASEGAWA, T., OGISHI, T., AND YAMAMOTO, S. AS alliance: Cooperatively improving resilience of intra-alliance communication. In *Proc. of ROADS Workshop* (December 2008).

[9] INTERNET EVOLUTION. Google Losing up to $1.65M a Day on YouTube. http://www.internetevolution.com/author.asp?section_id=715&doc_id=175123.

[10] JACOBSON, V., SMETTERS, D., THORNTON, J., PLASS, M., BRIGGS, N., AND BRAYNARD, R. Networking Named Content. In *Proceedings of ACM CoNEXT 2009* (Rome, Italy, Dec 2009).

[11] KATZ, D., AND WARD, D. Bidirectional forwarding detection. Internet-draft, 2009.

[12] KOPONEN, T., CHAWLA, N., CHUN, B.-G., ERMOLINSKIY, A., KIM, K. H., SHENKER, S., AND STOICA, I. A Data-Oriented (and Beyond) Network Architecture. In *Proceedings of ACM SIGCOMM 2007* (Kyoto, Japan, Aug 2007).

[13] KUMAR, K., AND SARAPH, G. End-to-End QoS in interdomain routing. In *Proc. of IEEE ICNS* (2006).

[14] KUSHMAN, N., KANDULA, S., KATABI, D., AND MAGGS, B. R-BGP: Staying connected in a connected world. In *4th USENIX Symposium on NSDI* (April 2007).

[15] LABOVITS, C., AHUJA, A., BOSE, A., AND JAHANIAN, F. Delayed internet routing convergence. In *Proc. of ACM SIGCOMM* (August 2000).

[16] REKHTER, Y., LI, T., AND HARES, S. A border gateway protocol 4 (BGP-4). RFC 4271, January 2006.

[17] WALTON, D., RETANA, A., CHEN, E., AND SCUDDER, J. Advertisement of multiple paths in BGP. Internet-draft, 2008.

[18] XIANGJIANG, H., PEIDONG, Z., KAIYU, C., AND ZHENGHU, G. AS alliance in inter-domain routing. In *Proc. of IEEE AINA* (March 2008).