Towards a Next Generation Inter-domain Routing Protocol

Lakshminarayanan Subramanian* Matthew Caesar* Cheng Tien Ee* Mark Handley†
Morley Mao‡ Scott Shenker*§ Ion Stoica*

1. INTRODUCTION

After a long period of neglect, there has been a recent resurgence of research on BGP, the current inter-domain routing protocol. Some of these papers have provided valuable empirical data on the current state of inter-domain routing [3, 16, 13, 12, 14, 7]; others have proposed incremental modifications that would improve the status quo [15, 4]. However, there has been a relative paucity of papers exploring how to fundamentally redesign inter-domain routing. In this paper we venture into this void, proposing a clean-sheet redesign of BGP. Our proposal is a hybrid link-state path-vector routing protocol, called HLP, which we offer not as the final word on inter-domain routing but rather as a possible starting point for debates about the future architecture of inter-domain routing.

There seems little disagreement that BGP is in need of eventual overhaul. In fact, the IRTF has convened two separate working groups to define the set of requirements for a future generation inter-domain routing protocol. From their combined set of specifications [9], we identified five problems of paramount importance, and describe the ways in which BGP fails to meet them:

Scalability: Any future inter-domain routing protocol must gracefully accommodate the ongoing growth of the Internet. BGP fails this test, as its routing state and rate of churn (the rate at which routing announcements are received by a given router) grow linearly with the size of the network.

Security: Given its critical role in today's telecommunication infrastructure, it is paramount that the Internet be robust, both to benign misconfigurations and to malicious attacks. Unfortunately, as recent Internet outages have made clear, BGP, by blindly accepting as valid the routing announcements of peers, is vulnerable on both counts; a single compromised or misconfigured router can cause extensive damage by propagating bogus route advertisements.

Convergence and Route Stability: To provide reliable reachability, Internet routes should be relatively stable and, when a change is necessary, the routes should quickly converge to their new steady-state. BGP, on the other hand, is known to suffer from significant route instabilities, route oscillations and long convergence times.

Isolation: Isolation is related to the three issues discussed above, but important enough to single out. No design can be robustly scalable if a single localized fault can impact the entire network. In BGP, unfortunately, changes in a single route are frequently propagated globally and many updates observed at a router are largely a result of events far removed from the router.

Diagnosis support: Routing protocols are designed to automatically adapt to faults, but they should also provide operators with enough information to quickly (and correctly) diagnose these faults, whether the cause is malicious or benign. BGP is notoriously deficient in this regard because the protocol conveys no information about the cause of a change or the intent of a peering. Also, the fact that a single failure or minor configuration change can be spread globally makes it difficult to localize the root-cause of routing problems.

This listing of BGP's flaws is hardly new, and several serious BGP flaws have already been dealt with by modest incremental modifications [15, 4, 19]. However, we contend that BGP's basic protocol structure makes it inherently incapable of achieving the aforementioned goals. We make this argument by discussing, at a general level, five basic design issues (Section 2). For each issue we review BGP's design choice, describe its impact on our goals, and then briefly describe HLP's approach. We then give a more comprehensive and detailed description of HLP (Section 3), and conclude (Section 4) with a few comments about open questions.

2. BASIC DESIGN ISSUES

We now describe five basic design issues that face any designer of inter-domain routing algorithms: routing structure, policy, routing granularity, routing style, and

*EECS Department, University of California at Berkeley. Email:{lakme,mccaesar,ct-ee,istoica}@eecs.berkeley.edu
†CS Department, University College, London. Email: mjh@cs.ucl.ac.uk
‡EECS Department, University of Michigan. Email:zmao@eecs.umich.edu
§ICSI center for Internet Research, Berkeley. Email:shenker@icsi.berkeley.edu

Table 1: Primary distinctions between HLP and BGP

Design issue	BGP	HLP
Routing structure	Flat	Hierarchical
Policy structure	Support for generic policies	Optimize for common case of policies
Granularity of routing	Prefix based	AS based
Style of routing	Path vector	Hybrid routing
Security and trust	No security, blind trust	verify correctness, minimize configuration errors

trust and security. This is not meant to be an exhaustive list, but is limited to the areas where (in our opinion) BGP is in most need of modification. For context, Table 1 summarizes the primary distinctions between HLP and BGP across the five design issues. There are two recurrent themes underlying these individual design differences. One is that, to simplify the design, we identify and optimize for the common cases. This applies to both policy and routing granularity. The other theme is to reduce interdependence by limiting the extent to which two ASs can affect each other.

We now discuss each of these five design issues individually. For each, we contrast BGP's approach with HLP's and describe how HLP overcomes the shortcomings of the current BGP design.

2.1 Routing Structure

The design of BGP assumes a flat routing structure, in which every AS treats every other AS equally and the protocol interactions between two ASs is agnostic of the type of relationship between them. As a result, the basic design does not specifically distinguish between different routing announcements which makes *local* routing events to be potentially *globally* visible [8]. This impairs BGP's scalability, and also makes it fundamentally hard to isolate routing events [9, 8]. Moreover, the resulting interdependence between ASs makes the entire Internet vulnerable to localized security or configuration problems; a single configuration error or compromised router can affect the rest of the network [14].

To reduce interdependence and, more specifically, to limit the extent to which route advertisements need to be propagated, HLP uses a hierarchical structure. Using a hierarchy, by itself, does not reduce interdependence. In HLP, we leverage the hierarchy to *hide* routing dynamics across nodes in different hierarchies to limit interdependence. There is no obvious pre-defined hierarchy one can leverage for this task, but we note that the typical relationships between interconnected ASs — peers, customers, and providers – defines a natural hierarchy, and this is what HLP uses. We now elaborate upon HLP's treatment of policy.

2.2 Policy

ASs, by their very autonomy, can have very different notions of what traffic they want to carry, and where they want their own traffic to be forwarded. Thus, one of the key differences between inter-domain and intra-domain routing is the need for such policy controls. BGP has a set of policy parameters that include export rules, import rules and local preferences. BGP is policy-neutral in that it attaches no semantics to policy parameters beyond their local implications for how to handle route advertisements. Moreover, it keeps the policy parameters fully private; only the resulting actions are visible, the underlying policies themselves are not.

This policy neutrality has allowed ASs great freedom in setting their policies. However, it has come at a cost, in that BGP is completely unable to distinguish between a misconfigured policy and a genuine one. This makes BGP much harder to manage and diagnose, and more susceptible to misconfigurations and attacks.

In weighing the benefits and costs of policy neutrality, we note that most ASs do not completely avail the policy flexibility at their disposal. In particular, the vast majority of relationships between ASs can be categorized as peers, customers, or providers, and over 99% of the policy settings follow two simple guidelines based on these categories [6, 18, 5]:

Export-rule guideline: Do not forward routes advertised by one peer or a provider to another peer or provider.[1].

Route preference guideline: Prefer customer-routes over routes advertised by peers or providers.

These rules are well-motivated by both economic and stability reasons. But the important consideration for HLP is that, if strictly adhered to by all ASs, these rules result in a strict hierarchical routing which follows provider-customer relationships. As mentioned above, hierarchical routing limits dependencies and thereby reduces routing churn and improves the extent to which faults can be isolated.

To take advantage of this in HLP, we explicitly publish the provider-customer relationships and restrict the normal set of available paths to a destination to those that obey this hierarchy. HLP does allow policies that do not obey these two simple rules, but it treats those as *exceptions* and provides additional mechanisms for supporting them. The result is a routing protocol that, in the common case, can recognize misconfigurations and limit the propagation of route advertisements. While one may think that publishing these AS relationships violates policy-privacy, most provider-customer relationships are inferable from BGP routing tables with a high degree of accuracy [18, 5]. Publishing these relationships does not reveal the financial terms of these relationships, nor does it reveal any exceptional policies.

[1]A specific variation to the export guideline which we do not consider as a violation is indirect-peering. Some ASs forward announcements from one peer to another peer either due to indirect peering (lack of direct connectivity) or due to sibling relationships (two AS's under same administration).

2.3 Routing Granularity

BGP uses prefix-based routing. While the initial design of BGP promoted aggregation of prefixes to improve scalability, what we notice today is the opposite phenomenon - route deaggregation for traffic engineering and policy routing. This in combination with the advent of many /24 networks has resulted in an alarming rise in the number of distinct prefixes in a routing table. Additionally, since BGP treats routes to each prefix in isolation, a single routing event triggers a separate routing update for each prefix. Moreover, even though routing is done at this fine granularity, the resulting routes mostly reflect the AS structure rather than the more detailed prefix structure. In HLP, we choose to separate routing from addressing by routing at the granularity of ASs rather than on prefixes; measurements suggest that at any given time, the number of distinct paths from a vantage point to the same destination AS is no more than 2 for more than 99% of ASs [2].

Given that prefix based routing results in greater churn and larger routing tables, and yet does not usually result in differing paths, HLP routes at the granularity of AS's instead of prefixes. This separates routing from addressing, which had been conflated in BGP. In addition to reduced routing state and churn, routing at the AS granularity has several ancillary benefits. Because the mapping between address prefixes and locations (as identified by AS) is much more static than the topology of the network, more appropriate transport and security mechanisms can be used for the topology information and for the AS-to-prefix mapping information. This, in turn, allows for easy detection of *origin misconfigurations*, in which an AS erroneously claims ownership of another AS's prefix.

2.4 Routing Style

BGP uses path-vector routing. Path-vector routing enables complex policies (since it enables ASs to base their policies on the entire path) and easy loop-suppression. But path-vector protocols, by revealing complete information along the path, have poor convergence properties. The worst-case convergence of a path-vector protocol is known to grow exponentially with the length of the path [11, 13]. Path vector routing also introduces unnecessary interdependence[2] which impedes the scalability and isolation properties of the protocol.

The alternatives to path-vector (PV) are the standard distance-vector (DV) and link-state (LS) styles of routing neither of which are good candidates for supporting policy based routing. DV routing does not reveal any information about the path to a destination and hence makes it fundamentally hard to apply policies on routes. LS routing, on the other hand, may violate privacy norms of policies by revealing every activity to all destination AS's. In contrast, path vector routing allows AS's to apply policies without providing complete visibility to the underlying events causing route updates.

Apart from policies, LS and DV routing have their own protocol strengths and limitations. LS routing has fast convergence and incurs low churn, the latter because updates are for link events, not routing changes (In PV and DV routing, one link event can cause many route changes). Moreover, fault diagnosis is easy with LS protocols, because it provides complete visibility into the current state of the network. However, global visibility is antithetical to both scalability and isolation.

DV routing, in contrast, can be adapted to provide good isolation (as we show later in Section 3, nodes can hide minor cost changes to isolate the effect of routing events), but the isolation comes at the price of reduced visibility.

None of these approaches are ideal solutions, so HLP uses a hybrid of link-state and path-vector routing. At first glance this might seem overly complex, but the hierarchical structure provides a natural way to decompose routing between the two styles; HLP uses link-state within a given hierarchy of AS's (as specified by provider-customer relationships) and uses path-vector between hierarchies. The link-state component improves convergence and so reduces churn within a hierarchy, while the path-vector component preserves global scalability by hiding internal route updates across hierarchies (thereby sacrificing global visibility). As such, HLP strives for a balance between visibility and isolation.

2.5 Security and Trust

The current BGP routing infrastructure is vulnerable to both accidental misconfigurations and deliberate attacks because BGP blindly trusts any route assertion from an authenticated router as valid[3]. This naive trust model stems from the lack of an overall security story (as, for instance, S-BGP would provide), and also from BGP's flat routing structure, where there are no semantic distinctions between classes of AS's. But it is clear that not all AS's are created equal; stub networks account for 85% of the AS's, and anecdotal evidence from ISPs suggests that a large fraction of these are poorly managed networks highly susceptible to configuration errors.

HLP addresses security in three ways. First, it does not treat all AS's as equal but instead limits the policy choices available to stub networks so that their misconfigurations cannot have substantial impact on the rest of the network. Second, HLP incorporates non-PKI security mechanisms, described in [19], which are incre-

[2]A single routing event on a link triggers route updates to every AS that utilizes some path traversing the link thereby making a large fraction of routing events to be globally visible.

[3]Just because a router is authenticated does not imply that it always propagates correct information. A router with a configuration error or compromised by an attacker can propagate bogus information.

mentally deployable as a first line of defense to help detect and mitigate misconfigurations and attacks. Third, the structure of the HLP hierarchy also allows it to easily adopt a PKI-based security mechanism similar to Secure-BGP [10]. While the conventional wisdom is that PKIs are hard to deploy, HLP's routing model is well-suited to a PKI because the certification hierarchy follows the pre-existing AS hierarchy of provider-customer business relationships. We only require the tier-1 ISPs to agree on a set of public keys to support this model; the public keys of each customer AS can be certified by its providers.

The discussion of these five design issues was intended to give a flavor of the intuition behind HLP's design. In the next section we describe how HLP actually works.

3. THE HLP ROUTING MODEL

In Figure 1 we show a sample AS-level topology consisting of several provider-customer AS hierarchies, each rooted at a tier-1 AS. A multi-homed AS can be part of more than one hierarchy. In this figure, each hierarchy is based only on the basic provider-customer relationships and does not incorporate any complex relationships (*e.g.,* two ISPs may have different relationships in different geographic regions) and those that an AS intends not to reveal. With HLP, we represent every such complex relationship as a peering link.

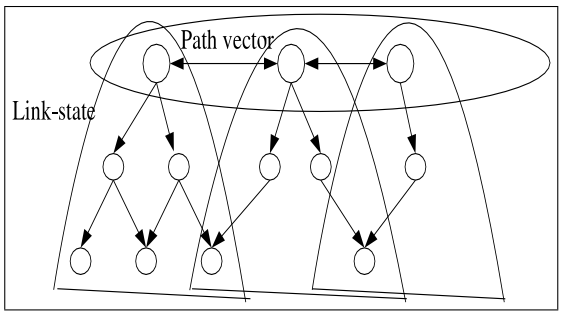

Figure 1: An AS hierarchy indicating provider-customer and peer-peer relationships

HLP uses link-state (LS) routing within each AS hierarchy and path-vector routing across peering links between AS hierarchies. Within a hierarchy, when an inter-AS routing event occurs, the other ASs in the hierarchy are notified using a link-state announcement. This link-state announcement is at the granularity of ASs and not at the granularity of routers. Every AS maintains link-state information about the inter-AS provider-customer links within its own hierarchy (inclusive of the links above it) and updates this information upon receipt of a link-state update.

Between hierarchies, the path-vector part of HLP is similar to BGP, where an AS propagates reachability information tagged with an AS path. The primary distinction is that the HLP uses a *fragmented path vector (FPV)* that contains only a portion of the AS path to the destination, rather than the entire AS path as with BGP. The FPV omits the portion of the AS path within an AS hierarchy. As the length of the FPV path has no routing significance, every FPV advertisement also carries a cost metric.

3.1 Basic Route Propagation Model

We now describe through example the basic model of how routes are propagated within and between AS hierarchies. Each node maintains a link-state topology database and a path-vector style routing table. Nodes exchange two types of messages: link-state advertisements (LSAs) and fragmented-path vectors (FPVs).

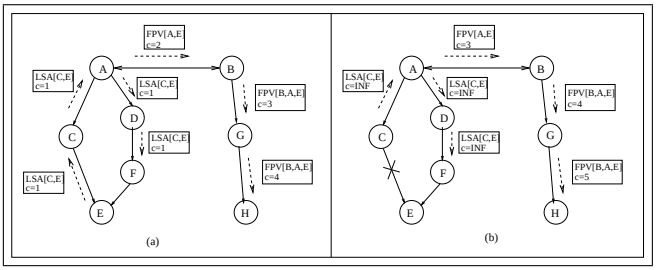

Figure 2: Basic HLP route propagation: Link failure example

Consider link (C, E) in Figure 2(a). Initially an LSA informs all the nodes in A's hierarchy of the existence and cost of link (C, E) (here, we consider all links to have a cost of 1). A receives the LSA, and propagates a path-vector to B, with FPV (A, E) and a cost metric of 2. The path vector is then distributed down the hierarchy to H without further modification of the path - neither the path within A's hierarchy nor the path within B's hierarchy appear in the FPV.

When link (C, E) subsequently fails (Figure 2(b)), nodes within A's hierarchy receive an LSA to inform them of the link-failure. Since A has an alternate path within its own hierarchy, A sends a path-vector update to B with a modified cost. This is essentially the same as a route withdrawal in BGP. In turn, B propagates the FPV down its own hierarchy to H. If however, A did not have an alternate path, A will propagate a route withdrawal to B.

FPV advertisements may be propagated across more than one peering link. Such forwarding allows HLP to express indirect peering, complex AS relationships, and sibling relationships where two ASs are under the same administration. In such cases, the FPV path includes all the peering ASs along all the paths to avoid routing loops or the need to perform a cost count to infinity.

To summarize HLP's basic routing model:

- All ASs maintain a link-state database of the topology in their local hierarchy.

- When an FPV is sent, the AS path in the FPV includes all ASs whose peering links were traversed, but excludes the parts of the path within the AS hierarchies.
- All inter-AS links have a cost metric which is added to the net cost value in an FPV route advertisement.
- HLP can model complex relationships by allowing the forwarding of route advertisements across more than one peering link.

Observation 1: *If every AS follows the HLP route propagation rules and every AS chooses a customer route if one exists, then the routing protocol is devoid of routing loops and the count to infinity problem.*

Moving from a complete path-vector protocol to a fragmented path-vector protocol does not introduce routing loops, nor does it require a count to infinity to remove information (as with Distance Vector protocols), *provided* every AS has an additional route selection rule to prefer a customer route if one exists. This additional constraint forces routes to follow the AS hierarchy, which is exactly the part of the path omitted from the FPV.

3.2 Hiding route changes using costs

The simple route propagation model above is functional, but is insufficient to achieve good scalability and isolation. To improve these two metrics, we use cost-hiding. The basic philosophy is to *propagate a route update only when necessary.* When an AS sees a cost increase or failure on the primary route to a destination, it checks if it has an alternate route with *comparable cost* to that of the previous route. If so, it suppresses the announcement to other ASs. The notion of *comparable cost* relaxes the notion of shortest path routing a little, and helps achieve better scalability and isolation.

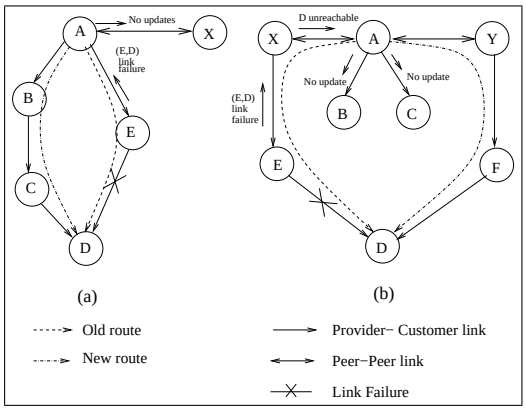

Figure 3: Two forms of cost-hiding. (a) AS A chooses an alternate route within its own hierarchy. (b) AS A chooses a route using an alternate peering link (A, Y) and hides the change from its customers.

There are three forms of cost-hiding in HLP: (a) not propagating minor changes across peering links; (b) not propagating minor changes to customers; (c) hiding the failure of one of multiple parallel peering links between a pair of ASs. The first two cases are illustrated in figure 3, and involve cost hiding by an AS higher up in the hierarchy that the origin of the change. In the third case, the issue is local to the two ASs, and it is entirely their own choice whether or not to advertise a cost change.

Observation 2: *If every AS chooses the customer route if one exists, then HLP with cost hiding is devoid of routing loops and the count to infinity problem.*

Routing loops and count to infinity problems still do not occur when AS's employ cost hiding, provided AS's explicitly choose customer routes as default. If they violate the default case, then this needs to be handled as an exception.

3.3 Handling exceptions

The HLP design is predicated on making the common case explicit. However, we still need to handle exceptions to the common case. For example, in the case of a backup link, an ISP may prefer to route through a peer or a provider rather than directly to a customer. We solve these problems by degrading gracefully towards the BGP path vector design. In the extreme scenario, when every route is an exception, the routing dynamics of HLP degenerate to those of BGP, but still maintain the advantages of the separation of addressing and routing. The basic approach for supporting exceptions is to treat them like a peering link, and hence apply FPV across these links. To illustrate this, we discuss three examples of exception handling.

Exception 1: Choosing Non-Customer Routes.

An AS X that prefers to choose a non-customer route over a customer route, performs two operations. First, X propagates an exception to all its providers and peers withdrawing its customer route. Second, X propagates an FPV corresponding to the chosen non-provider customer route to its customers. In essence, these operations are equivalent to executing HLP in the case where the customer did not exist in X's hierarchy.

Exception 2: Forwarding from a Provider to a Peer.

To violate the AS hierarchy and forward a route from a provider to a peer, an AS treats the provider-customer link as a peering link. Hence, it first converts an LSA received from the provider to an FPV containing the provider-customer link and propagates this FPV to a neighboring peer. This translates to the case of having an FPV traverse multiple peering links.

Exception 3: Forwarding from a Peer to a Provider.

Similar to the previous exception, forwarding an announcement from a peer to a provider translates to treating the customer-provider link as a peering link. Hence, an FPV announcement from a peer will be propagated to the provider with the path-vector in the FPV including all the three AS's involved in the exception.

To summarize exception handling: *any network that*

chooses to forward a route in violation of the constraints on a provider-customer link should model the link as a peering link (with regards to this route) and use the normal HLP propagation rules.

3.4 Preliminary evaluation results

In this section, we briefly summarize some of our preliminary evaluation results comparing HLP's performance to BGP and also describe our HLP implementation.

Scalability and Isolation: Based on analyzing the current Internet topology gathered from RIPE [17] and RouteViews [20], we found that under the idealized scenario where all AS's use the common case of policies, cost hiding in HLP: (a) reduces the churn rate roughly by a factor of 50 in comparison to BGP; (b) isolates the location of a fault to a region roughly 20 times smaller than that of BGP. Additionally, using AS-based as opposed to prefix-based routing provides a 7.8 factor reduction in churn.

Convergence time: Labovitz *et al.* [11] has shown that the worst case convergence time of BGP in an n-node fully connected graph is $O((n-1)!)$. In comparison, HLP achieves *linear time convergence in the common case* which covers a large fraction of routing events. This result stems from two observations. First, link-state routing within a hierarchy has linear-time convergence. Second, the path-vector convergence time varies as $O(n^k)$ where k is the length of the fragmented path-vector and for a large fraction of paths, $k = 1$ (for paths that traverse indirect peering links, $k = 2$).

HLP Implementation: Currently we have implemented HLP on top of XORP 1.0 [1], a software router platform. Our implementation reuses much of the code from XORP's BGP module. Apart from the basic design of HLP, our implementation can handle different forms of exceptions and also has inbuilt support for bootstrapping new routers into the network.

4. DISCUSSION AND CONCLUSIONS

In this paper, we provide the design of HLP as one concrete suggestion in the design space of future inter-domain routing protocols. HLP addresses five pressing problems of BGP - scalability, isolation, security, convergence and diagnosis. We hope our work stimulates discussion around two topics:

Future Internet architecture: Determining the right operational model for the future Internet architecture is the critical factor influencing the structure of next generation routing. HLP completely retains the operational and economic model of BGP but only alters the route propagation model of BGP. Other radically different Internet architectures like NIRA [21] and feedback-based routing [22] assume a very different underlying operational model than what is today. NIRA advocates better end-host control over routing while feedback-based routing computes source routes based on measurement feedback from the network.

Policy structure: There exists no manual to clearly outline the comprehensive suite of policies an ISP requires. The current set of policy practices may not be completely representative since many policies have evolved around the structure of BGP as opposed to being a basic feature. Designing a protocol to satisfy a policy suite without sacrificing the basic properties of a protocol is a challenging task.

5. REFERENCES

[1] The eXtensible Open Router Platform (xorp). http://www.xorp.org.

[2] D.-F. Chang, R. Govindan, and J. Heidemann. The temporal and toplogical characteristics of BGP path changes. In *Proc. International Conference on Network Protocols*, 2003.

[3] N. Feamster, D. G. Andersen, H. Balakrishnan, and M. F. Kaashoek. Measuring the effects of internet path faults on reactive routing. In *Proc. ACM SIGMETRICS*, 2003.

[4] N. Feamster, J. Borkenhagen, and J. Rexford. Guidelines for interdomain traffic engineering. *ACM Computer Communication Review*, 2003.

[5] L. Gao. On inferring automonous system relationships in the internet. *IEEE/ACM Trans. Networking*, to appear 2004.

[6] L. Gao and J. Rexford. Stable Internet routing without global coordination. In *Proc. ACM SIGMETRICS*, 2001.

[7] R. Govindan and A. Reddy. An analysis of Internet inter-domain topology and route stability. In *Proc. IEEE INFOCOM*, 1997.

[8] T. Griffin. What is the Sound of One Route Flapping?, 2002. IPAM talk.

[9] IRTF Routing Research Group. http://www.irtf.org/rrg/.

[10] S. Kent, C. Lynn, and K. Seo. Secure Border Gateway Protocol (Secure-BGP). *IEEE Journal on Selected Areas of Communications*, 18(4):582–592, April 2000.

[11] C. Labovitz, A. Ahuja, A. Bose, and F. Jahanian. Delayed Internet Routing Convergence. In *Proc. ACM SIGCOMM*, 2000.

[12] C. Labovitz, A. Ahuja, and F. Jahanian. Experimental study of Internet stability and wide-area network failures. In *Proc. International Symposium on Fault-Tolerant Computing*, 1999.

[13] C. Labovitz, R. Malan, and F. Jahanian. Origins of Internet routing instability. In *Proc. IEEE INFOCOM*, 1999.

[14] R. Mahajan, D. Wetherall, and T. Anderson. Understanding BGP Misconfigurations. In *Proceedings of ACM SIGCOMM*, 2002.

[15] Z. M. Mao, R. Govindan, G. Varghese, and R. Katz. Route flap damping exacerbates Internet routing convergence. In *Proc. ACM SIGCOMM*, 2002.

[16] J. Rexford, J. Wang, Z. Xiao, and Y. Zhang. Bgp routing stability of popular destinations. In *Proc. ACM Internet Measurement Workshop*, 2002.

[17] RIPE's Routing Information Service Raw Data Page. http://data.ris.ripe.net/.

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

[19] L. Subramanian, V. Roth, I. Stoica, S. Shenker, and R. Katz. Listen and Whisper: Security Mechanisms in BGP. In *Proceedings of ACM/ USENIX NSDI*, 2004.

[20] University of Oregon RouteViews project. http://www.routeviews.org/.

[21] X. Yang. Nira: A new internet routing architecture. *ACM SIGCOMM FDNA Workshop*, 2003.

[22] D. Zhu, M. Gritter, and D. Cheriton. Feedback based routing. *ACM Hotnets Workshop*, 2002.