

Stabilizing BGP, Safely

P. Brighten Godfrey, Matthew Caesar, Ian Haken, Scott Shenker, Ion Stoica
{pbg,shenker,istoica}@cs.berkeley.edu, mccaesar@gmail.com, haken@berkeley.edu

Abstract

Route instability is widely recognized as a major problem in the Internet. Core routers are barraged with millions of updates daily, leading to massive infrastructural costs and worsened data-plane performance. Route flap damping provides some protection against instability, but introduces pathologies and reduces availability.

With concerns about the scalability of the routing system prompting a renewed interest in stability, we believe it's time for a more principled approach to stabilizing Internet routing. This paper takes a step towards that goal by characterizing the tradeoff between stability and availability. In a large-scale simulation of the BGP protocol supplied with traces of measured inter-AS adjacency failures, we bound the performance of theoretically optimal strategies and evaluate the performance of several implementable strategies. Motivated by the principle that any improvements to stability should not come at the price of availability, we argue for an approach which we call *Stable Route Selection* (SRS). Our numerical evaluation shows that SRS preserves the high availability of BGP without flap damping, while obtaining stability similar to BGP with flap damping. Although further evaluation is necessary, these results indicate a promising approach to safely stabilizing BGP.

1 Introduction

A number of studies point to stability as a key problem for the Border Gateway Protocol (BGP), the interdomain routing protocol which knits the fabric of today's Internet [9, 14, 25]. Network failures, policy changes, and the BGP convergence process itself can generate huge numbers of routing updates, causing problems in both the data and control planes.

In the data plane, it is well known that end-to-end path quality is degraded by BGP route updates (see [26] and references therein). According to a recent study, the majority of packet loss bursts are caused by inter-domain route convergence problems such as transient forwarding loops, rather than by congestion [25]. These problems are increasingly important as the Internet is becoming an ubiquitous platform for voice and video applications. For example, in the Skype VoIP client—which has over 3.5 million simultaneous active users [6]—delays of more than 250ms or the loss of more than three consecutive packets are enough to interfere with speech compre-

hension [6]. Internet games such as Counter-Strike have similar demands for interactivity, commonly sending periodic delay-sensitive bursts of packets [8].

In the control plane, a storm of route updates can overload routers, which leads to increased processing delays, increased route convergence time, and packet loss. A study of the Sprint network found that BGP processes consumed the majority of CPU cycles on core routers [1]. Based on measurements of a Telstra core router, Huston [13] calculated that update processing consumed an average of 30% of a 1.5GHz processor, with peak load higher. As the routing table size is increasing faster than Moore's law, instability translates into increasingly more expensive routers and higher convergence times [15]. These problems led the Internet Architecture Board Workshop on Routing and Addressing to recently identify update churn as one of the challenges for future scalability of the routing system [18].

The main mechanism for improving stability in BGP is route flap damping (RFD) [21], an heuristic which filters routes that have a short-term update rate above some threshold. However, this seemingly simple approach is fraught with problems. In 2002, Mao et al. [17] demonstrated that flap damping creates pathological conditions that slow route convergence. Flap damping also worsens *availability*—the fraction of time that a router has a route to a destination—by occasionally shutting off *all* available routes. The operator community has become aware of these problems, with the RIPE Route Working Group advising in 2006 that “the application of flap damping in ISP networks is NOT recommended. ... With current vendor implementations, BGP flap damping is harmful to the reachability of prefixes across the Internet.” [20] Other approaches to improve stability have required protocol modifications [5, 17] or have addressed narrow cases such as eliminating persistent oscillations arising from dispute wheels [10, 11] or selecting routes at a multihomed edge host [2]. Recently, Li and Huston [16] proposed several new heuristics as RFD replacements, but their performance and possible side-effects have not been fully evaluated.

With concerns about the scalability of the routing system prompting a renewed interest in stability [16, 18], we believe it's time for a more principled approach to stabilizing Internet routing. In this paper, we take a step towards that goal by exploring the tradeoff between stabil-

ity and availability. First, we give techniques for bounding the performance of *theoretically optimal strategies*, which require knowledge of the future but allow us to constrain which points in the tradeoff space are achievable, for any given network topology and pattern of failures. We apply these techniques to a large-scale simulation of the BGP protocol in an environment based on Internet measurements. Second, we evaluate the performance of *implementable strategies* in this environment.

This characterization of the tradeoff space leads us to argue for a new approach which we call *Stable Route Selection* (SRS). SRS is motivated by the principle that any improvements to stability should not come at the price of availability. Rather than *shutting off* unstable routes as in flap damping, SRS *prefers* more stable paths over less stable paths, when multiple options are available. Intuitively, compared with an approach which shuts off unstable routes, SRS’s stability will be at least as good as long as at least one stable route is available; and if all paths are unstable, SRS may have worse stability but will have better availability.

Our numerical evaluation shows that SRS preserves the high availability of BGP without flap damping, while obtaining $5\times$ better stability—slightly better than BGP *with* flap damping. Moreover, the heuristic we propose comes within $1.6\times$ of the theoretically optimal stability for any availability-preserving strategy, even if knowledge of the future is allowed.

We acknowledge two areas in which further work is required. First, our simulations of the BGP control plane use a metric for stability which is a proxy for concrete metrics of interest such as router CPU utilization and packet loss in the data plane; experimental results on real or software routers could measure these quantities directly. Second, our dataset has notable deficiencies: for example, we have only partial knowledge of ISP routing policies, and our characterization of the tradeoff space should be tested on a broader range of scenarios to see how our conclusions generalize. However, we believe the above results, based on one year of data from Route Views [29], indicate a promising approach to safely stabilizing BGP.

The rest of this paper proceeds as follows. Section 2 gives a classification of approaches to stabilizing BGP, describes how we bound the performance of theoretically optimal policies in each class, and describes the SRS strategy. In Section 3, we describe our evaluation environment, and give the numerical results in Section 4. We conclude in Section 5.

2 Stabilizing Internet routing

We begin this section with a formalization of our main metrics, availability and stability (Section 2.1). We next give a classification of approaches to stabilizing BGP (Section 2.2) and describe how to bound the maximum

possible improvement for each approach (Section 2.3). We then give some background about the standard BGP decision process (Section 2.4) and how we modify it for our scheme, SRS (Section 2.5).

2.1 Metrics

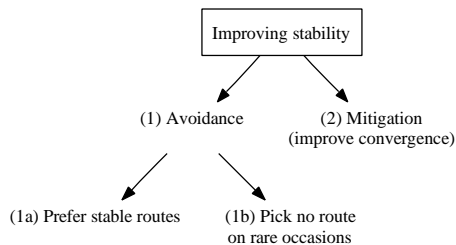
We define *availability* for a particular source-destination pair as the mean fraction of time that the source has a route to the destination. We will typically study the mean availability over all source-destination pairs.

To measure stability, we use *interruption rate*: the number of times the selected path from the local AS to the destination changes or is withdrawn entirely (i.e., a transition to the disconnected state). We do not count recovery events as an interruption. We use interruption rate as a proxy for data plane performance and control plane CPU utilization due to its computational and analytical tractability.

2.2 Classification of approaches

For any particular source-destination pair, the interruption rate I can be seen as the product of the mean rate T of “external trigger” events which force a switch in paths, such as link failures or ISP policy changes; and the mean number C of interruptions per trigger: that is, $I = T \cdot C$.

To improve stability, we can thus either (1) *avoid* external triggers by reducing T , or (2) *mitigate the impact* of these events by reducing C , i.e., improving the re-convergence process. Within the avoidance approach, two ways to avoid external triggers are (1a) preferring paths that fail less often, or (1b) occasionally disconnecting the source from the destination. Presumably, occasionally disconnecting the nodes would be desirable only in extreme cases in which the only available paths are highly unstable, and avoiding the resulting control-plane processing becomes more important than reachability. Our classification is thus as follows:



As an example, flap damping’s policy—shutting off very unstable routes—combines techniques (1a) and (1b). When there is a choice between an unstable route and stable routes, if RFD damps the unstable route only, it has the effect of preferring the more stable path. When *all* available routes are deemed unstable by RFD, it has the effect of shutting off all routes, thus disconnecting the node.

We wish to study each category separately, because they imply different fundamental tradeoffs. In particular,

occasionally picking no route reduces availability; preferring more stable routes implies giving less weight to path length or other objectives; and improving the convergence process may not have either tradeoff.

2.3 Bounding the optimal policies

In this section we outline how we bound the maximum possible improvement that can be obtained from the above approaches to improving stability in BGP. These procedures assume a given AS-level topology annotated with customer/provider/peer business relationships between ISPs, and a pattern of AS adjacency (“link”) failures.

We break down our description into three steps: factoring out convergence; picking optimally stable routes; and optimally trading off availability for stability. We implement these procedures using a modified version of our event-based BGP simulator (see Section 3) with appropriate pre- and post-processing steps. Due to space constraints, we only sketch the procedure, and omit formal proofs of optimality.

Convergence: In order to tractably provide a guarantee of optimality, our method requires a simplification of the environment: we assume that BGP update messages propagate instantly and are handled instantly at each router. This separates time into discrete *batches*, each of which is composed of one or more link state changes followed by a sequence of route changes until the network converges. Thus, at all points in time except during the instantaneous convergence process, each router has a valid path to the destination, or no path. (In our simulator, we implement this in such a way that the BGP update messages are processed in the same order that they would have been, had link delays been turned on and subsequent topology changes been delayed until after the convergence process completed.) Although batching alters the convergence properties of the system, we will observe similar performance with and without batching in Section 4.

Batching allows us to easily lower-bound the number of interruptions due to convergence. Given a fixed setting of each router’s converged state before and after each batch, there must be at least 1 interruption for each batch in which the route changed or was withdrawn, and at least 0 if the path stayed the same. To see how much improvement is possible by optimizing the convergence process, one can compare the this lower bound with the actual number of interruptions observed in the simulation.

Picking stable routes: We next describe how we select routes to minimize the total number of interruptions experienced by all nodes in the network, subject to the standard constraints imposed by customer/provider/peer business relationships. Consider first a single node minimizing its *own* interruption rate by selecting from among a given sequence of available routes R_1, \dots, R_m where each R_t is

the set of routes available at time t . In this case the minimum interruption rate can be obtained by a “*single-node optimal*” strategy: stick with the current route as long as it is available, and otherwise pick the route $r \in R_t$ which will be continuously available farthest into the future [12].

However, there are two complications stemming from the fact that nodes’ available options (R_t) are dependent on other nodes’ selections. First, the future failure time of a route is dependent on future changes in the business class of the route selected by each AS on the path. For example, if a downstream AS switches from a customer route to a peer route, it will no longer export the route to other peers or providers. We calculate business class switch times by running the route selection simulation twice, recording the business class switch times on the first trial, and using them to compute paths’ future failure times in the second. It can be shown, along the lines of the proof of convergence in [10], that the business class switch times will be identical in both trials.

The second complication is that even if we know the future failure time of a route, the single-node optimal procedure is not always optimal for the network as a whole, since one node’s selections affect others’ options. In fact, it is easy to construct examples wherein the optimal choice at a node has dependencies with nonlocal parts of the network. To deal with this issue, we allow each node to independently select its route, thus possibly picking a route which was not chosen by the downstream ASs along the path (but which still conforms to the business relationship constraints). This yields a lower bound on the optimal mean interruption rate.

Trading availability for stability: To produce optimal points in the availability-stability tradeoff space, we introduce a parameter λ which intuitively sets the cost of being disconnected per unit time. For any fixed λ , we will find the interruption rate, i , and amount of time spent disconnected, d , which minimize the objective function $f_\lambda(i, d) = i + \lambda \cdot d$. It is easy to see that the result is an undominated point in the (i, d) tradeoff space, and we can produce multiple points on the optimal tradeoff curve by varying λ . Moreover, since f is linear, we can optimize f for each node separately, as follows. From the previous step, we can collect, for each node and each time t , the valid route which will be available farthest into the future at time t . We then reduce the problem to a shortest paths computation by constructing a graph whose nodes represent possible routes or no route, and whose edges represent valid transitions between these states over time. Edges representing an interruption have cost 1, and edges representing t units of downtime have cost λt . A shortest path in this graph thus corresponds to a sequence of route selections which minimizes f_λ .

2.4 The BGP decision process

BGP affords a high degree of flexibility through the use of a *decision process*, which allows operators to customize selection to conform to goals such as traffic engineering or economic relationships. The BGP decision process consists of the following sequence of steps, which select a route based on *attributes* contained in the BGP route announcement: (1) Highest local preference (2) Lowest AS path length (3) Lowest origin type (4) Lowest MED (5) eBGP over iBGP-learned (6) Lowest IGP cost (7) Lowest router ID. The output of each step is a *set* of routes that are *equally good* according to that and every previous step. By adding, modifying, or filtering attributes in update messages, operators can control the specific route selected to reach a particular destination.

In our evaluation, since policies on the Internet are not widely disclosed, we will leverage [22] to infer and assign local preferences associated with business relationships as done in [14, 17, 23]. Since our simulator models each AS as a single router, we also simplify steps 3-7 of the process by assigning each AS a (uniform random) identifier, and selecting the route with the lowest next-hop ID, not unlike step 7. These simplifications of the process preserve what is, for our purposes, the most salient feature: preferences in the decision process are not correlated with stability of the paths.

BGP implementations commonly include an implementation of Route Flap Damping (RFD). RFD was designed to improve stability by holding down unstable routes. In particular, the router maintains a numeric penalty value $p_{P,N}$ associated with every (prefix P , neighbor N) pair. Upon receipt of an advertisement or withdrawal, the router increases $p_{P,N}$. When $p_{P,N}$ increases beyond a *cut-off threshold*, the route is excluded from consideration when selecting routes. The penalty decays exponentially, and the route is reconsidered for use when its value falls below a *reuse* threshold. Commonly used default settings for these parameters are given in [17].

2.5 Stable Route Selection

A stable route selection policy uses some of the flexibility in route selection to prefer more stable paths, thus targeting approach (1a) in the classification of Section 2.2. In the design evaluated in this paper, we accomplish this by inserting an additional step in the BGP decision process that prefers routes which (according to an heuristic) are less likely to fail soon. Specifically, we will insert SRS as the second step, thus preserving business relationship policies but trading off path length for stability.

The heuristic we insert is as follows:

1. Prefer the current route if it is still available.
2. Prefer routes with lowest AS path length.
3. Prefer the route with longest uptime, i.e., that has been advertised for the longest continuous length of time.

The “longest uptime” strategy has been used in many contexts (see [12] and references therein) and performs well since past behavior is frequently correlated with future behavior.

3 Evaluation environment

Data sets: We infer the inter-domain AS-level topology by culling AS adjacencies from Route Views [29] feeds and using [22] to characterize links as either provider-customer or peer-peer. We assume ASes distribute routes according to the common-case import and export policies as discussed in [22]. To infer the pattern of failures, we record the appearances and departures of links from the Route Views feeds. Specifically, we consider a link to be available at time t if some route which uses the link is currently advertised to a Route Views peer at time t . In this manner, we infer a trace of link state changes from Route Views from January 1, 2006, to December 30, 2006, which we replay against our simulator.

Simulator: To evaluate the performance of various route selection strategies, we use an event-driven BGP simulator extended from the simulator used in [7]. The simulator’s events are at the level of link state changes and BGP updates. Each AS is represented by a single node running a BGP instance, as in some past studies [4, 23]. Our simulator runs a simplified version of the BGP protocol described in RFC 1771 [19], as well as the decision process as in [21] and flap damping as in RFC 2439 [24]. Flap damping parameters were set according to Cisco default settings [17]. Inter-AS packet propagation delay is selected randomly for each packet, uniformly distributed between 5 and 15 ms.

In each trial we select a single random destination to which all nodes route over a random month of our year-long data. Each plot incorporates measurements of 500 trials. In each trial, we gather measurements only after the first 24 hours of simulated time, to eliminate initial convergence effects. Since some data is missing from our topology causing a minority of nodes to be always disconnected, when collecting measurements we ignore source-destination pairs whose availability in the Standard BGP strategy (without RFD) is < 0.99 . (This, however, did not substantially affect our results.)

4 Results

Figure 1 summarizes the main results of our evaluation in the stability-availability space. The next two sections discuss these results, comparing Standard BGP, RFD, and SRS (Section 4.1), and comparing against the theoretically optimal policies (Section 4.2). We then discuss SRS’s tradeoff with other objectives, in particular path length (Section 4.3).

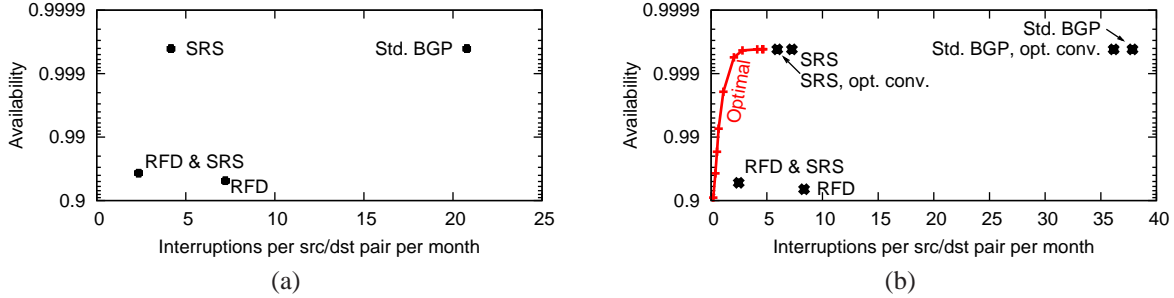


Figure 1: Performance of various strategies in the stability-availability space, with batching off (a) and on (b).

4.1 Comparison of Std. BGP, RFD, and SRS

We begin by comparing the points in Figure 1(a), where batching is disabled (that is, link delays and MRAI timers are turned on). Standard BGP maintains a high availability of 99.98%, but suffers from a high rate of 20.8 interruptions per month. Route Flap Damping (RFD) reduces the mean rate of interruptions by a factor of 2.9, but sacrifices two “nines” of availability. One might expect the tradeoff between these metrics to be fundamental. However, we can see that SRS is able to achieve the high availability of standard BGP with the low interruption rate of RFD. Using RFD and SRS in conjunction results in an additional 3.1-fold decrease in interruption rate over RFD and slightly improves availability. This is to be expected, since by picking more stable paths, RFD is triggered less often.

Figure 2 explores the pattern of interruptions in more detail with a CCDF over all measured end-to-end paths. Standard BGP’s long tail shows what other studies [9] have observed: a small number of Internet routes suffer from persistent instability. Both SRS and RFD drastically reduce the size of this tail. SRS is able to achieve roughly the same benefit in the tail as RFD without incurring RFD’s reduction in availability. Interestingly, and unlike RFD, SRS is able to improve the stability for the upper part of the curve, i.e., for routes that have only moderate instability. Finally, when we combine SRS with RFD, we note that the instability of the most unstable routes is reduced by almost an order of magnitude, while only incurring the availability reduction of RFD in isolation.

4.2 Comparison with optimal policies

Figure 1(b) compares the strategies in the batched environment (see Sec. 2.3) wherein we can lower-bound the interruption rate of the optimal policies. We see a substantially similar relationship between the strategies with batching on and off, which suggests that batching is a reasonable approximation under which to compare strategies. The main difference is inflated interruption rates, which can be explained by the fact that in Fig. 1(a) some link state changes may be effectively skipped as a result of link delays and MRAI timers, while in the batched environment the system finishes reconverging after every topol-

ogy change.

Figure 1(b) also bounds the benefit possible under the three general techniques classified in Section 2.2: improving convergence, preferring stable routes, and accepting some downtime. First, comparing the points for Standard BGP and SRS with their hypothetical counterparts with optimal convergence—which transition from the initial to the final path in each batch without any path exploration process—shows that in our environment, convergence has only a minor contribution to interruption rate. We stress, however, that it is possible that real-world scenarios would have greater convergence overhead. For example, ISPs may withdraw prefixes either intentionally or due to configuration error, triggering long sequences of path hunting which are relatively rare in our environment since we only model link state changes.

Second, the figure shows that SRS performs close to optimal among strategies which prefer more stable paths (without sacrificing availability), coming within a factor 1.6 of optimal, or just $1.27\times$ if convergence is factored out. These results indicate that our heuristic does an effective job of predicting the relative stability of paths.

Finally, the “Optimal” curve in Figure 1(b) demonstrates limits on how much improvement can be gained by occasionally disconnecting nodes. This lower bound admits the possibility that stability can be improved to about 2 interruptions per month with small availability loss, but any further improvements come at the cost of significantly more downtime. For example, reaching 1 interruption per month requires reducing availability from 99.96% to below 99.8%, i.e., over $4\times$ as much downtime.

4.3 Tradeoffs with other objectives

SRS requires flexibility in route selection in order to prefer more stable paths. That flexibility is also important to optimize other objectives, such as preferring routes based on business relationships (which we have respected in the results presented here), minimizing path length, or other policies. Unfortunately, it is difficult to obtain measurements which characterize, for real-world ISP policies, how much flexibility would be available to SRS after more important objectives are satisfied. However, we can char-

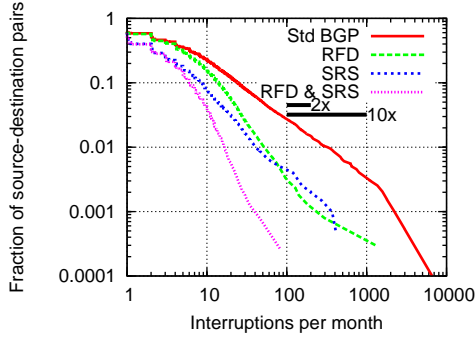


Figure 2: Complementary CDF of interruptions per month over all measured source-destination pairs.

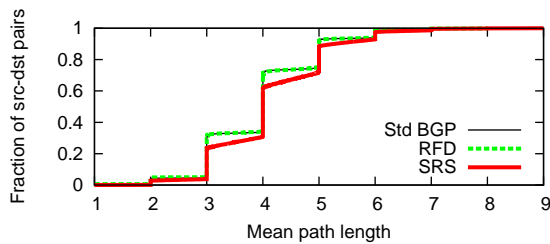


Figure 3: CDF of mean path length over all measured source-destination pairs.

acterize the tradeoff with one such objective, AS-level path length.

Figure 3 shows the CDF of mean path length over source-destination pairs. SRS has a mean path length of 4.14 AS-level hops, or just 4.2% greater than Standard BGP’s 3.97 hops. We note that BGP itself empirically incurs an AS-level path inflation of 49.8% itself due to policies [3], so our inflation of may be tolerable.

This low inflation may be expected since part of the SRS heuristic prefers shorter paths (Sec. 2.5). In addition, path lengths in this environment are constrained by the hierarchical nature of the AS graph when business relationships are satisfied: we found that a hypothetical strategy which always preferred *longest* paths would have mean path length just 32% longer than Standard BGP.

A more challenging objective to deal with is load balance: an ISP might wish to balance its traffic evenly across links, rather than preferring stable paths. We envision two potential solutions. First, the ISP may use an SRS-like strategy which prefers more stable paths, but which eventually returns to the load-balanced paths after they have remained stable for some time. Second, if the ISP has the ability to route multiple classes of traffic along different routes, it would be possible to send the most stability-sensitive flows (*e.g.*, real-time voice traffic) along SRS paths, and other flows along whichever paths minimize maximum link utilization.

5 Conclusion

In this paper, we have explored the space of techniques for improving stability in BGP by characterizing the tradeoff between availability and stability. Our early results indicate that *Stable Route Selection*—preferring stable paths, rather than shutting off unstable paths—is a promising strategy for mitigating instability in Internet routing without sacrificing availability.

There are several important areas of future work. We are currently building and deploying a prototype implementation of SRS based on the Quagga [28] open source router, as an way to validate and extend our simulation results. Additionally, the evaluation we have presented here could be enhanced greatly by a quantitative characterization of how much flexibility could be available to SRS under real-world ISP policies. Finally, it may be interesting to apply SRS to stabilize intra-domain routing, perhaps by adjusting link weights to reflect historical stability.

References

- [1] S. Agarwal, C. Chuah, S. Bhattacharyya, C. Diot. “Impact of BGP dynamics on router CPU utilization,” Passive and Active Measurement Workshop, April 2004.
- [2] A. Akella, J. Pang, B. Maggs, S. Seshan, A. Shaikh, “A Comparison of Overlay Routing and Multihoming Route Control,” ACM SIGCOMM 2004, Portland, OR.
- [3] M. Caesar, T. Condie, J. Kannan, K. Lakshminarayanan, I. Stoica, S. Shenker, “ROFL: routing on flat labels,” ACM SIGCOMM, Sept. 2006
- [4] H. Chan, D. Dash, A. Perrig, H. Zhang, “Modeling adoptability of secure BGP protocols,” ACM SIGCOMM, Sept. 2006.
- [5] J. Chandrashekar, Z. Duan, J. Krasky and Z. Zhang, “Limiting Path Exploration in BGP,” INFOCOM, March 2005
- [6] K. Chen, C. Huang, P. Huang, C. Lei, “Quantifying Skype user satisfaction,” ACM SIGCOMM, September 2006.
- [7] C-T. Ee, V. Ramachandran, B-G. Chun, K. Lakshminarayanan, and S. Shenker, “Resolving inter-domain policy disputes,” ACM SIGCOMM, August 2007.
- [8] W. Feng, F. Chang, W. Feng, J. Walpole, “Provisioning On-line Games: A Traffic Analysis of a Busy Counter-Strike Server,” Internet Measurement Workshop, November 2002.
- [9] A. Feldmann, O. Maennel, Z. Mao, A. Berger, B. Maggs, “Locating Internet routing instabilities,” ACM SIGCOMM, August 2004.
- [10] L. Gao, J. Rexford. “Stable Internet Routing without Global Coordination,” IEEE/ACM Transactions On Networking, 9(6):681–692, December 2001.
- [11] T. Griffin, F. Shepherd, and G. Wilfong, “The Stable Paths Problem and Interdomain Routing,” IEEE Transactions on Networking, Volume 10, Issue 2 (April 2002).
- [12] P. B. Godfrey, S. Shenker, and I. Stoica, “Minimizing churn in distributed systems,” ACM SIGCOMM, Sept. 2006.
- [13] G. Huston, “2005—A BGP Year in Review.” 21st APNIC Open Policy Meeting, February 2006.
- [14] C. Labovitz, A. Ahuja, A. Bose, F. Jahanian, “Delayed internet routing convergence,” Proc. ACM SIGCOMM, 2000.
- [15] T. Li. “Router Scalability and Moore’s Law,” Workshop on Routing and Addressing, Internet Architecture Board, October 2006.
- [16] T. Li and G. Huston. “BGP Stability Improvements.” Internet-Draft, June 13, 2007. *draft-li-bgp-stability-01.txt*
- [17] Z. Mao, R. Govindan, G. Varghese, R. Katz. “Route Flap Damping Exacerbates Internet Routing Convergence,” ACM SIGCOMM, August 2002.
- [18] D. Meyer, L. Zhang, and K. Fall. “Report from the IAB Workshop on Routing and Addressing.” Internet-Draft, April 13, 2007. *draft-iab-raws-report-02.txt*
- [19] Y. Rekhter and T. Li., “A Border Gateway Protocol 4 (BGP-4), RFC1771,” March 1995.

- [20] P. Smith and C. Panigl. "RIPE Routing Working Group Recommendations on Route-flap Damping" Document ID ripe-378, May, 2006.
- [21] J. Stewart, "BGP4: inter-domain routing in the Internet," Addison-Wesley, New York, 1999.
- [22] L. Subramanian, S. Agarwal, J. Rexford, R. Katz, "Characterizing the Internet Hierarchy from Multiple Vantage Points," in *IEEE Infocom 2002*, June 2002.
- [23] L. Subramanian, M. Caesar, C. Ee, M. Handley, M. Mao, S. Shenker, I. Stoica, "HLP: A Next-generation Interdomain Routing Protocol," ACM SIGCOMM, August 2005.
- [24] C. Villamizar, R. Chandra, R. Govindan, "BGP route flap damping, RFC2439," Nov 1998.
- [25] F. Wang, N. Feamster, L. Gao, "Quantifying the effects of routing dynamics on end-to-end Internet Path Failures," Technical report TR-05-CSE-03, University of Massachusetts.
- [26] 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," ACM SIGCOMM, Sept. 2006.
- [27] Abilene Observatory Data Collections, <http://abilene.internet2.edu/observatory/>
- [28] Quagga software routing suite. <http://quagga.net>
- [29] "Route Views Project," <http://routeviews.org>.
- [30] C. Panigl, J. Schmitz, P. Smith, C. Vistoli, "RIPE Routing-WG Recommendations for Coordinated Route-flap Damping Parameters." <http://ripe.net/docs/ripe-229.html>