The Case for an Internet Health Monitoring System

Matthew Caesar, Lakshminarayanan Subramanian, Randy H. Katz
{mccaesar,lakme,randy}@cs.berkeley.edu

Abstract

Internet routing is plagued with several problems today, including chronic instabilities, convergence problems, and misconfiguration of routers. We believe that a first step towards making the Internet robust to these problems is by developing a systematic methodology for analyzing routing changes and inferring *why* they happen and *where* they originate. In this paper, we motivate the need as well as describe the design of an Internet health monitoring system that identifies the source of routing instabilities purely by passively observing routing updates from different vantage points. We believe such a system could be used to continuously infer the state of the network. Such inferences may then be used *offline* for network performance monitoring and troubleshooting, or *online* to improve path selection and damping of instability.

1 Introduction

Diagnosing Internet routing problems is a highly challenging task for network operators. On many occasions, locating the source of a problem and fixing it requires the collective effort of several operators in different administrative domains. Unfortunately, the current diagnosis methodologies used by operators are often very ad-hoc and time-consuming processes involving phone calls and email exchanges [13].

Compounding this problem is the fact that the Border Gateway Protocol (BGP), the de facto inter-domain routing protocol, has evolved into a complex protocol with a number of configurable policies, making the dynamics of Internet routing hard to model and comprehend. BGP operates at the granularity of Autonomous Systems (ASes) where operators in each AS independently make policy decisions. This makes it difficult for operators to predict the impact of simple configuration changes on global routing dynamics. Hence, existing efforts to diagnose problems in BGP and to modify the protocol to eliminate its shortcomings have become a black art.

To assist in diagnosis, this paper proposes the design of an *Internet health monitoring system* that analyzes route updates from multiple vantage points and localizes the source and potential cause of routing events that triggered each update. Although pinpointing the cause of certain routing changes may be fundamentally hard [5], for certain types of massive routing events (*e.g.,*, session resets), it is often possible to pinpoint the exact inter-AS link which might have underwent the reset due to the large number of constituent updates [17]. We believe such a system can be useful for network operators in several ways. First, sharing health inferences across administrative domains can reduce operator time spent investigating problems and speed their repair. Second, creating a repository for historical information about observed events using the inferences from our system can be used by operators to set policies. Third, the current mechanisms used for damping route flaps (to improve stability) is known to exacerbate route convergence [7]. Using a health inference system could enable application of damping selectively at a much finer granularity. Finally, overlay networks like RON [2] or route control devices [1] can utilize these inferences to selectively probe and choose alternative paths that have better stability properties.

We have developed a full-fledged health monitoring system that has been operational for over 18 months. During this period, we have been able to pinpoint several routing anomalies of high magnitude many of which were previously unknown. We found a number of inter-AS links to be perennially unstable and found certain links to be particularly prone to failure when the Internet was congested (e.g., when Internet worms were propagating). While our proposed design currently uses only passive BGP update information, we believe that the precision can be enhanced by coupling it with active probing diagnostic tools like AS-traceroute [8].

2 Types of routing problems

BGP's rich set of configurable policies and cross-protocol interactions make it highly subject to a wide range of routing problems. We illustrate three very important problems that occur on a regular basis today:

Continuously flapping routes: BGP is known to have poor convergence properties [7]. A *continuously flapping* route is a sign of an extreme convergence problem where a prefix is repeatedly updated over long periods of time. Flapping is clearly undesirable since it affects route availability as well as places a phenomenal load on routers. We found that routes to a surprisingly large fraction of prefixes (25%) are involved in a continuously flapping event at any point in time. Moreover, some flapping events would last for long periods of time, for example, one such event lasted for 80 days.

High magnitude events: Some BGP events simultaneously affect a large number of routes within a short period of time. For example, a single session between two BGP routers that resets can simultaneously affect route availability for nearly $150,000$ prefixes as well as trigger a separate route convergence process for each prefix. Session failures occur on a daily basis and form the primary reason for massive bursts of updates and routing changes in the Internet. They can be triggered by flash crowds and worm outbreaks, which can cause the TCP channel carrying the BGP session to suffer significant packet loss. We observed over $25,573$ session resets during our 18 month period. However, we found that most resets were caused by a small number of sessions: the 9% most unstable sessions contributed 49% of the total number of observed resets.

Misconfigurations: Nearly $200 - 1200$ prefixes are affected by misconfigurations every day [9]. There are three key types of misconfigurations. (1) *address hijacking:* where an AS announces one or more prefixes belonging to another AS. Certain hijacking events are hard to protect against and have caused large outages in the Internet on several occasions. (2) *policy violations:* where the AS advertises a route that is in violation of its policy. For example, routes from providers and peers are typically not forwarded to other providers and peers, as there is no economic incentive for an ISP to forward such traffic. These incidents typically do not cause connectivity problems but they do bring more traffic into the AS, which can trigger congestion [9]. (3) *leaking of routes:* where routes that should be filtered due to configuration or aggregation rules are unintentionally leaked to neighbors. Leaking of routes can increase load on routers and routing table size, and can trigger router reboots if the routing table exceeds memory capacity [11]. One key class of leaked routes are advertised prefixes that are a subblock of a larger, more persistently available prefix. We found that routes in this class are particularly unstable, with 50% of these routes being withdrawn within 8 minutes of being advertised.

3 Challenges to diagnosis

In this section we discuss three key challenges to diagnosis in BGP and describe ways we aim to address them.

AS as a distributed system: Each AS is a large network comprising of thousands of routers. BGP advertisements at the AS-level provide zero visibility into intra-AS routing dynamics. Hence it is hard to determine whether a problem exists within an AS or on inter-AS peering links. Moreover, since ASes may be connected by multiple peering links, two AS-level paths that intersect in the AS-topology might not intersect in the network-level topology, as shown in Figure 1. We address these issues by designing inference mechanisms that do not make assumptions about network level topology.

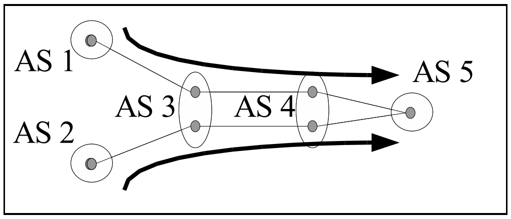

Figure 1: Two paths that intersect in the AS-level topology might not intersect in the router-level topology. Hence an event taking place in the intersection of the two AS-level paths might not affect both routes.

Simultaneous events: To determine the cause of a routing event, it is necessary to determine the set of updates triggered by the event. However, given the vast size of the Internet and the high rate of routing events, multiple routing events may simultaneously affect routes to the same prefix. This may cause route advertisements triggered by the two events to overlap in time. Hence, it becomes a difficult task to determine when one can safely determine whether a group of routing updates for a prefix is triggered by a single event.

We address this by separating prefixes into two groups: *stable* prefixes that are rarely updated, and *continuously flapping* prefixes that are repeatedly updated over long periods of time. For stable prefixes, events are separated by long periods of time, making separation of updates tractable. Most popular prefixes (prefixes that sink the most traffic) also tend to be very stable, and hence events affecting data traffic are more likely to be observable. Separating the updates triggered by different events for a continuously flapping prefix is fundamentally hard.

Visibility of events: Not all events may cause an observable route update in BGP. There are two key reasons why updates from an event might not be observed: (i) The state of the route might change through several intermediate states while the route is being filtered or dampened. (ii) The event may not affect any of the views due to its location. In addition, minor events may be unobservable in the presence of major events, since the few updates caused by a minor event get subsumed as noise when a major event triggers many route updates at the same time. However, if a minor event affects a prefix that is not continuously flapping, then updates to it are typically visible.

We found that if an event affects a prefix that is not continuously flapping, then updates to it are typically visible. Moreover, we found that an event with a large magnitude is visible at a vantage point provided the vantage point has a large number of routes traversing the location where the event occurred. In addition, the observed magnitude of an event is fairly constant regardless of distance (AS hop count) from the view. To increase the chances we observe an event, we use multiple vantage points. In practice, we found that using $3-4$ vantage points is sufficient to observe most minor events.

4 Diagnosis approach

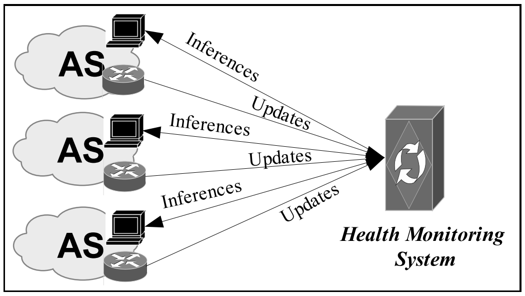

Figure 2: Architecture of health monitoring system.

The health monitoring system (Figure 2) collects updates from multiple vantage points (routers) located in various ASes, whose owners volunteer to make their routing information available to the health monitor. The monitor, in turn, reports its inferences to each vantage point. Routeviews [15] and RIPE [14] are two-real world examples of update collection systems where one can deploy a health monitor. In this section we provide a brief overview of the system.

We define a *routing event* as an activity taking place at some location in the network that generates one or more route updates. A route update observed by the health monitoring system is a clear signal of some routing event in the Internet. BGP route updates are at the granularity of prefixes and every update contains an ASPATH attribute which describes the entire routing path at the AS level to the destination prefix. This ASPATH attribute is the primary *clue* at the disposal of a health monitor to determine the location and potential cause of routing events.

4.1 System components

Our health monitoring system consists of three components. First, the *Local-view inference engine* collects routing updates via BGP peering sessions with routers. Its role is to infer the location of events purely based on local observations. Next, the *Aggregator* aggregates the potentially large volume of inferences from each view and determines a refined set of events that might have potentially occurred. Finally, the *Health monitor* computes different statistics on the inferences, and triggers alarms when *unhealthy* behavior is detected (*e.g.,* events of high magnitude, recurrent events etc.). This component also provides an interface for applications to query statistics or set traps for various alarms. We continuously publish the results from our health monitor based on Routeviews data on a web site [18].

Separating the functionality of our system into these components provides several benefits. First, information is aggregated and refined locally before sending them to the health monitor, thereby reducing bandwidth requirements and improving scalability (especially given that these components may be geographically distributed). Second, we obtain flexibility in deployment where an operator may choose to deploy a set of components within their own AS for internal monitoring, without joining the centralized health monitor. Third, information considered proprietary can be filtered or anonymized between components (especially given that operators may wish to restrict information flow across AS boundaries due to privacy considerations). Finally, components can be replicated and appropriately placed to improve fault-tolerance and availability during failures.

4.2 Inference methodology

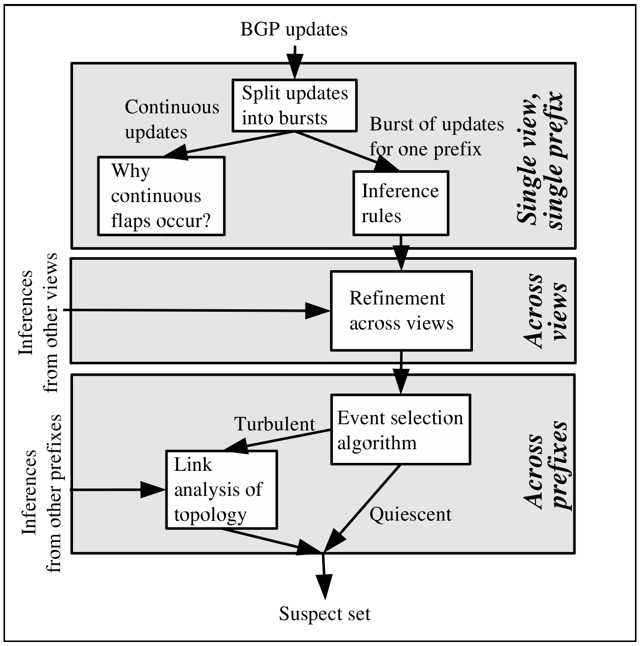
BGP updates
Split updates into bursts
Continuous updates
Burst of updates for one prefix
Why continuous flaps occur?
Inference rules
Single view, single prefix
Inferences from other views
Refinement across views
Across views
Inferences from other prefixes
Turbulent
Event selection algorithm
Link analysis of topology
Quiescent
Across prefixes
Suspect set

Figure 3: Algorithm fbw.

The primary design challenge in building a health monitoring system is the methodology used for inferring routing events purely based on passively observing route updates. In particular, the goal is to determine the set of locations where the event could have occurred, and for each location a list of possible causes that could have occurred at that location.

Given the complicated nature of this problem and the associated challenges, we split the problem into different subproblems and tackle each in isolation, as shown in Figure 3. We only touch upon the salient aspects of our methodology in this section (refer to [17] for details). Previous work [4] [3] has demonstrated the ability to localize events based solely on BGP updates. Our inference methodology extends previous work along three dimensions:

Separating stable from continuously flapping prefixes: We separate prefixes which are relatively stable from those that get continuously updated. We observe that for stable prefixes, two properties commonly hold: (a) routing events affecting these prefixes are typically visible and not affected by damping; (b) two different events affecting the same prefix are separable in time, making it possible to distinguish updates from different events. These two properties make root-cause analysis for these prefixes feasible. Root-cause analysis is more challenging for continuously flapping prefixes, and so we use a different methodology for these prefixes as discussed in our technical report [17].

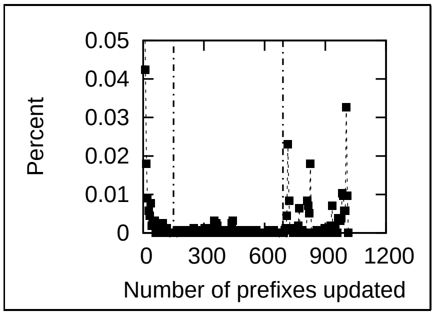

Figure 4: Frequency of time intervals with various numbers of updated prefixes for a view in AS 1239. We use the gap (separation shown with vertical dashed lines) to distinguish Turbulent from Quiescent periods. Less than 0.2% of the mass is within the gap. We found similarly large gaps for other links and views.

Improve vs. Worsen classes: While the list of causes of routing events is innumerable, many of these causes can be categorized into two classes: events that cause the path to *Improve*, and those that cause the path to *Worsen* according to the BGP path selection process. For each update, our system determines which of the two classes each AS in the AS path falls into, giving rough information about the cause of the event. We can also intersect observations across views and prefixes for each class to further narrow down location. As future work, we aim to define a more rich set of classes, to obtain a finer distinction between different types of causes.

For example, suppose a view V observes a single route change from path P_B to P_A to a destination D as shown in Figure 5. While one can associate several causes for this route change, we can definitely conclude that the routing event either *improves* the properties of P_A causing it to be the more preferred path or *worsens* the properties of P_B by making it less preferred. Hence, the underlying cause of the event, though unknown, can be classified into the improve or worsen categories. If the event occurred along P_A, then the cause belongs to the *Improve* class, otherwise it belongs to the *Worsen* class if it occurred in path P_B. Knowing the classes can also help narrow down location: since an event must be in the same class regardless of view, our system intersects ASes appearing in the Improve (i.e. ASes along path P_A) and Worsen (ASes along P_B) classes across views. Thus far, we have defined only a small set of classes and defining a more complete set to obtain a finer distinction between different types of causes is a subject of future work.

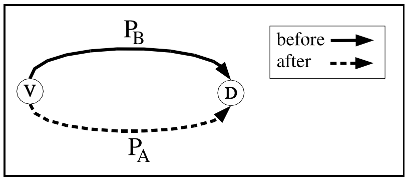

Figure 5: Example: A single prefix undergoes an ASPATH change.

Correlation of routing updates: One of the key problems we need to address is how we correlate updates that are caused by the same event without incorrectly correlating observations from different events. We correlate updates across three dimensions: *time*, *prefixes* and *views*. From a single view, we cluster updates to each prefix based on how close they occur together in time so as to separate routing updates triggered by different routing events. Since the prefix is not continuously flapping, bursts of updates are likely to be caused by the same event. Next, we cluster across prefixes by noting that a large number of prefixes simultaneously updated in a short period of time can indicate that the multiple prefixes are affected by the same event. We found that time intervals can be clearly classified into *Quiescent* and *Turbulent* periods based on the number of prefixes updated during that time (Figure 4), and show that many observations in a Turbulent period are triggered by the same event.

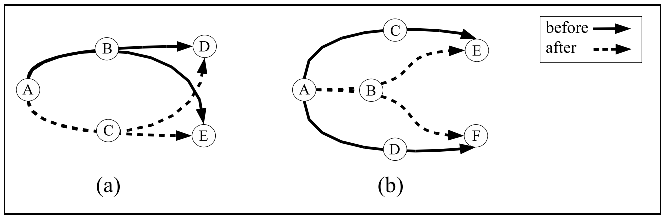

Figure 6: Example: Effects of a MED decrease on a single view.

For example, suppose A uses B to reach D and E, as shown in Figure 6(a). Suppose many prefixes simultaneously change to start using C. Given that such a large burst of updates is rare, it is likely they were all caused by the same event. Moreover, we can determine the event most likely occurred on the common sub-path across all these prefixes. Hence there are two possible causes: either an event took place on (A, B) that worsened the properties of the path, or an event took place on (A, C) that improved the properties of the path. A second example is shown in Figure 6(b). Suppose many prefixes simultaneously change to start using (A, B). Since it is unlikely two simultaneous major events occurred on both $[A, C, E]$ and $[A, D, F]$, it is likely that a single major event either (a) took place at (A, B) that improved the properties of the path, or (b) took place internally in A that worsened the paths of several prefixes using (A, C, E) and (A, D, F) (since A is the only common AS across the sub-paths). In both these examples, we can separate ASes into *improve* and *worsen* categories and perform Turbulent inference on each class independently.

5 Implementation results

We have built a prototype of our system and used it to analyze route route updates collected from Routeviews and RIPE over a period of 18 months. We describe some preliminary observations that we believe show such a system can both be used by network operators to isolate and repair faults, and also to provide better insight into Internet routing dynamics arising from those faults.

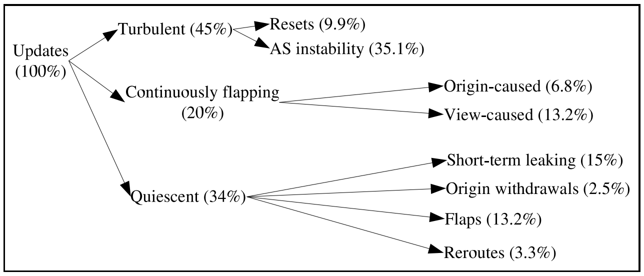

Figure 7: Breakdown of updates by cause.

Although we were unable to pinpoint the cause for any update, we were able to narrow down the cause into a several categories. Figure 7 gives the number of updates for each type of event. We found that we could pinpoint the location of virtually all major events we detected, and could narrow down the location of Quiescent events to within 2 ASes for 20% of updates. We found the majority of continuously flapping prefixes could be pinpointed to a single AS. Overall, we could pinpoint the location to a single inter-AS link (pair of ASes) for 70% of updates.

Previously unnoticed events: A number of important routing anomalies go unnoticed in the Internet today. Our system was able to detect many events which have gone unnoticed in the past. We give three examples here. First, on January 2002, a Chinese ISP underwent on average two session resets per day for a period of two weeks, affecting reachability to over 1000 prefixes in China. Second, in July 2003, the peering link between AS 3561 and AS 1239 underwent a large number of session-reset like events, affecting the reachability of over 20,000 prefixes including the domains cnet.com, excite.com, and weather.com. Third, in June 2004, AS 2500 began to advertise paths for over 500 prefixes it did not own, affecting reachability to several major providers.

Continuously flapping routes: From our analysis of BGP updates, we detected two distinct categories of flapping prefixes: (a) a prefix is classified as continuously flapping by several views (*near-origin* flaps); (b) a prefix continuously flaps only with respect to a single view (*near-view* flaps). Many near-origin flaps are triggered due to the widespread deployment of route control products [1] for traffic engineering purposes, while many near-view flaps are caused by instabilities within the AS containing the view. In practice, we found that near-origin flaps are much more likely to affect reachability than near-view flaps. Also, we found that a small fraction of the prefixes (1%) are particularly prone to continuously flapping events, and thereby trigger a disproportionately large number (90%) of routing changes.

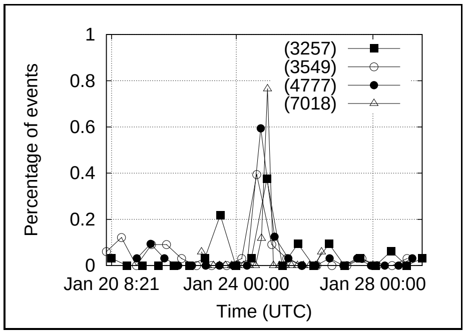

Figure 8: Effect of the SQL slammer worm on the number of major events detected (we filter out effects of local resets). Each line corresponds to observations made at a different vantage point.

High magnitude events: We observed 39,387 large events (events affecting more than 1000 prefixes) during the 18 month period. Roughly two thirds of these appeared to affect inter-AS adjacencies while the rest appeared to be due to instability inside an AS. We found certain links to be perennially unstable, and we found certain links to be highly prone to instability: over 50% of large events occurred within 2 hours of another large event affecting the same link/AS. Although large events are more rare in the Internet core (tier-1 ISPs) than at the edges, we found the rate at which these events occur in the core increases by an order of magnitude when Internet worms are propagating. For example, Figure 8 shows a sharp increase in the number of major events taking place while the SQL slammer worm spreads.

Correctness: Although validation can be difficult because ISPs are (understandably) unwilling to share information about faults occurring inside their networks, we use several approaches to validate the correctness of our inferences. First, we compare our inferences on certain route updates where the cause and location are well known. These updates may have been manually injected as part of experiments [6] or they may be due to large historical events reported on in the media. Next, we find large events in our inferences and perform post-mortem analysis to attempt to explain their cause [16]. Then we cross-validate inferences made at each view in isolation and check for conflicts. Finally, we correlate our observations with logs from two large ISPs with cases when we infer that ISP to be the cause based on external observations. In each of these cases, we found our inferences to match with what was known to be correct.

6 Conclusions and research agenda

This paper has made the case that building a health monitoring system for the Internet is feasible. We presented some preliminary measurements from a prototype implementation that suggest such a system could benefit network operators and end users. However, there is substantial future work to be done to improve upon the system design. First, we aim to improve inference accuracy, by leveraging active measurements, and by developing a methodology to determine which parts of the network are most critical for deploying vantage points. Second, we plan to develop a more robust distributed implementation that can handle the heavy update loads of the Internet core. We plan to complete the design of the health monitor, by developing guidelines for determining which observations constitute unhealthy behavior, so system can then trigger alarms when these observations occur. Finally, demonstrating feasibility of this system opens the door to several interesting avenues of future work, including how the system can be used to simplify network management and troubleshooting, and how the inferences can be used in an online fashion to improve stability of Internet and overlay routing.

References

[1] A. Akella, J. Pang, B. Maggs, S. Seshan and A. Shaikh, “A Comparison of Overlay Routing and Multihoming Route Control”, ACM SIGCOMM 2004.

[2] D. Andersen, H. Balakrishnan, M. Kaashoek, R. Morris, “Resilient overlay networks,” in Proc. of SOSP, Banff, Canada, October 2001.

[3] D. Chang, R. Govindan, J. Heidemann, “The Temporal and Topological Characteristics of BGP Path Changes,” in *Proc. of ICNP*, Atlanta, GA, November 2003.

[4] A. Feldmann, O. Maennel, Z. Mao, A. Berger, B. Maggs, “Locating Internet routing instabilities,” in *Proc. of SIGCOMM*, Portland, Oregon, August 2004.

[5] T. Griffin, “What is the sound of one route flapping?,” presentation made at the *Network Modeling and Simulation Summer Workshop*, 2002.

[6] Z. Mao, R. Bush, T. Griffin, M. Roughan, “BGP beacons,” in *Proc. Internet Measurement Conference*, October 2003.

[7] Z. Mao, R. Govindan, D. Varghese, R. Katz, “Route flap damping exacerbates Internet routing convergence,” in *Proc. of SIGCOMM*, Pittsburgh, PA, August 2002.

[8] Z. Mao, J. Rexford, J. Wang, R. Katz, “Towards an accurate AS-level traceroute tool,” in *Proc. of SIGCOMM*, Karlsruhe, Germany, August 2003.

[9] R. Mahajan, D. Wetherall, T. Anderson, “Understanding BGP misconfiguration,” in *Proc. of SIGCOMM*, Pittsburgh, Pennsylvania, August 2002.

[10] Y. Rekhter, T. Li, “A Border Gateway Protocol 4 (BGP-4)”, IETF, *RFC 1771*, March 1995. Section 9, pgs 33-46

[11] L. Wang, X. Zhao, D. Pei, R. Bush, D. Massey, A. Mankin, S. Wu, L. Zhang, “Observation and analysis of BGP behavior under stress,” in IMW, November 2002.

[12] C. Villamizar, R. Chandra, R. Govindan, “BGP route flap damping,” RFC 2439, November 1998.

[13] “NANOG Mailing List,” `http://www.merit.edu/mail.archives/nanog/`

[14] “RIPE RIS,” `http://www.ripe.net/ris/`.

[15] “Route Views Project,” `http://www.routeviews.org`.

[16] “Sprintlink: Scheduled Maintenence and Outage” `http://www.sprintlink.net/maintview/`

[17] M. Caesar, L. Subramanian, R. Katz, ”Towards a BGP health inferencing system,” U.C. Berkeley Technical Report UCB/CSD-04-1321, November 2004.

[18] `http://www.cs.berkeley.edu/˜mccaesar/hmon.html`