Internet Routing Anomaly Detection and Visualization

Tina Wong Van Jacobson Cengiz Alaettinoglu

Abstract

Some researchers have speculated that Internet's routing system would experience a collapse in the near future. Although this has yet to be proven, diagnosing inter-domain routing problems is hard. BGP, the defacto inter-domain glue, is designed for routing, not diagnosis. It is extremely chatty — the most minor connectivity change produces hundreds of BGP messages and a major peering loss can generate millions — and making sense of the deluge of data remains challenging. We have developed statistical algorithms to extract the large-scale structure of BGP events and visualization techniques to display that structure in operationally meaningful ways. These tools can be used to detect routing anomalies in real-time on a modern processor. We also describe how to integrate additional data sources into the algorithms to combat the drawbacks of using BGP events alone and to better understand issues in inter-domain routing. We show case studies of routing instabilities automatically diagnosed by these tools from the viewpoints of a Tier-1 ISP and a large institutional network.

I. Introduction

The Internet comprises of Autonomous Systems (ASs) and relies on two-level routing: *intra*-AS and *inter*-AS. An AS is a collection of network resources under the administrative control of a single entity. An AS can use any intra-AS routing protocol (IGP). ISPs usually run OSPF or ISIS as their IGPs, whereas enterprises commonly run EIGRP or OSPF. IGP can only route data from one point to another inside an AS; it does not know routes to external ASes. Instead, an AS runs BGP [1], the defacto inter-AS (or inter-domain) routing glue, to facilitate connectivity to other ASs. A BGP router peers with routers in other ASes, and sends route announcement messages containing reachability information to prefixes either originating from its own AS or from ASs it is willing to transport traffic. It also sends withdrawals when routes become unreachable or infeasible.

In inter-domain routing, ASes can have different roles. An enterprise AS should not allow outside networks to use itself as transit, while an ISP's business model is to provide that service. An AS achieves this differentiation through configuring routing policies. As BGP route announcement messages move along the Internet, routers attach to them path attributes: AS paths to denote the ASes traversed; community tags to distinguish routes from one peer from another; multi-exit-discriminator (MED) to express a metric of preference when two ASes connect to each other in more than one

place. Depending on its role, an AS configures policies based on these attributes on its BGP routers to filter or allow routes or carry out specific actions to achieve its business goals.

Although the protocol itself is simple, the routing policies associated with BGP are complex. Various intricate Internet routing behavior are realized through configuring policies on many routers. The process is intensive and error-prone. Serious misconfiguration incidents have put BGP on news headlines when main parts of the Internet went down because of routing anomalies. Once, a small AS accidentally announced the full Internet routing table with one-hop AS paths, and most ASes started to prefer those routes because of the very short paths. Unintentionally, the small AS became transit for majority of Internet traffic. It was not able to handle the avalanche, taking down the Internet with it for more than a day. This sort of catastrophe has been rare. A more common, day-to-day anomaly is route hijacking, in which a BGP router announces reachability to prefixes it does not own nor able to route to, in effect black-holing the traffic to those prefixes. Route hijacking is usually the result of misconfigurations, for example, a typo advertising the prefix 1.3.3.0/24 instead of 1.2.3.0/24. But it can also be malicious, employed as a form of DoS attack. Another anomaly is route leakage, when a misconfigured BGP router mistakenly sends a lot of routes to a peer with limited resources, which can completely melt down the peer router. In one such incident, ISP-A's [1] customer leaked thousands of extra routes to ISP-A. ISP-A unwittingly announced those routes to its peer in ISP-B. ISP-B had a max-prefix-limit configured just for this sort of occurrence, so that its routers would not be overwhelmed. The BGP session closed, severing the communication between ISP-A and ISP-B — they can no longer send traffic to one another. Some BGP dynamics also severely affect routing stability, for example, the well-known persistent MED route oscillation problem [2]. Because of the way MED is used in the BGP route selection algorithm, there can be a lack of total ordering among a set of available routes to a prefix, causing a router to alternate its preference between these routes to the prefix, persistently, until intervention. This oscillation not only affects traffic for the prefix involved, but also more importantly, the router itself. The router can become so busy processing route announcements and withdrawals for that one prefix, its CPU utilization pushed to near 100%. Crucial real-time functions on the router can suffer as a result, such as responding to IGP keepalives from other routers required to maintain stable IGP routing.

In short, BGP can cause harm in multiple ways. Hence, carefully managing BGP is very important. Unfortunately, it is hard. The challenges are three-folds. First, BGP is designed to facilitate routing, not diagnosis. Take a peering session reset between two external BGP peers x and y for example. x is required by the protocol to explicitly withdraw from all

[1] We anonymized the names of the actual ISPs to protect their identities.

its other peers P all routes it heard from y and had announced to P. In turn, every peer in P withdraw those routes from its own peers, and so on, propagating to faraway places. The same goes for y and its peers and their peers. When the session is re-established, x and y exchange full routing tables with each other. Then, x announces the routes it previously withdrawn to P, and each peer in P does the same to its own peers, and so on. Likewise for y. BGP does not send out a message saying "peering between x and y reset". The peers of x and y do not know about a reset except the massive route withdrawals followed by their re-announcements. This is related to the second challenge of BGP — it is a very chatty protocol. Even for a simple path failover, the simplicity of BGP's path vector nature can trigger a long exploration process in which an AS tries all invalid AS paths before convergence. The third challenge is the sheer number of prefixes in BGP and the multiple possible paths to get to a destination. There are about 150K prefixes in the Internet. For a dual-homed AS, this would be 300K routes (i.e., paths to the prefixes); for an ISP, there could be millions of routes. One can issue a "show ip bgp" command on a router to look at its set of routes, but this would only show the current routes of that router, and there can be many routers in a network with frequently changing routing. Even on a single router, there can be a large number of routes — differences among routes might be subtle, but important, such as MEDs, but comparing them is not straightforward.

Finding out what is happening in the Internet through BGP is no easy task. While gigahertz processors and terabyte disks have made it possible to capture and record BGP messages via passive peering, making sense of the deluge of data remains difficult. In this paper, we tackle the challenges outlined above and present solutions to detect and visualize Internet routing anomalies. The major contributions are:

- "One picture says 1,000,000 routes": A visualization technique that shows the large-scale structure of some set of BGP routes in an operationally meaningful way. It can either draw a picture to present BGP routes at a point in time, or generate an animation to track routing changes in BGP events.
- An analysis technique that does anomaly detection with BGP events. It allows network operators to answer questions like "what happened during this upsurge of updates?", "where in the network did it happen?" and "how does it affect me?" in real-time on a modern processor. The technique is fast enough even when dealing with the entire backbone mesh of a typical Tier-1 ISP.
- We describe the integration of router configuration files, traffic data and IGP routing data into our algorithms to better understand Internet routing anomalies.

- We present Case studies on Internet routing instabilities, observed from within a large educational network and a major U.S. Tier-1 ISP. We present a number of hard-to-detect, problematic BGP incidents, automatically diagnosed by the above techniques. The incidents include backdoor routes, misconfigured community tags, policy filters with unintended consequences, unexpected leaked paths, persistent route oscillations, and peering traffic imbalance.

Roadmap. We first describe our data collection methodology in Section II. Section III goes over our algorithms for Internet routing anomalies detection and visualization, and describes the integration of additional data sources. Section IV presents results of case studies of real networks using the algorithms. We discuss related work in Section VI and conclude in Section VII.

II. Data Collection Methodology

We collected both BGP and IGP routing data from real networks using the Packet Design Route Explorer (REX) [3]. REX passively IBGP peers with all BGP edge routers at a site, or core route reflectors of an ISP, just like interior routers would IBGP peer with each other, i.e., our view of the BGP information is the same as other members of the site's IBGP mesh. For the data used in this paper, the routers passed REX their full routes. The BGP UPDATE messages by themselves are not sufficient for analysis because route withdrawals do not contain the attributes being withdrawn. We remedy this by maintaining an AdjRibIn for each of its peers. When a peer sends REX an explicit withdrawal or an announcement that implicitly invalidates a route, the peer's AdjRibIn tells us the original route attributes. Our data consists of BGP messages augmented with the withdrawn attributes. We call this an event stream or just events. REX also maintains an adjacency passively with a IGP router, or multiple adjacencies for a multi-area network, to collect IGP link state advertisements.

Our first dataset was collected at U.C. Berkeley from August to December 2003. Berkeley runs a four-area OSPF as its IGP. REX peered with four BGP edge routers. In early August 2003, REX saw 13 BGP Nexthops, approximately 12,600 prefixes and 23,000 routes at Berkeley. Berkeley's upstream provider CalREN was undergoing significant transitions at the time. CalREN was migrating its members to a CENIC backbone, and also was in the process of consolidating 5 AS numbers into 2, one for commodity and reliable access called Digital California, and another one for a high performance research network. As a result of this transition, BGP-related incidents happened more often than during the usual operating environment. Nonetheless, we believe most observations presented here are not limited to such drastic periods.

We also collected data at a U.S. Tier-1 ISP (ISP-Anon) from June to August 2002. To protect this ISP's identity we anonymize all IP addresses, prefixes, router names and AS numbers from the dataset. ISP-Anon runs ISIS as its IGP. REX peered with the full route reflector mesh with 67 route reflectors. In late June 2002, REX saw about 9150 different BGP Nexthops, 316 Originators, 1.5 million routes, 200,000 prefixes, and 850 neighbor ASes.

Although in this paper we only present IBGP data collected from inside an administrative domain, our algorithms are general and designed to apply to EBGP as well. Most BGP studies so far have focused on globally visible problems using data from publicly available servers such as RouteViews. We argue that diagnosis from within a site — a single AS, enterprise with multiple ASes,or ISP — is as critical if not more so, as a network operators cares most about effects on his own network and what he can do locally to fix a problem. As far as we know, this paper is the first to study root cause diagnosis and show BGP misbehaviors from within a network's point-of-view.

III. Algorithms

In this section, we present two algorithms that help detect and visualize Internet inter-domain routing anomalies. First, *TAMP* is a visualization technique that shows the large-scale structure of some set of BGP routes. Second, *Stemming* is an analysis technique that do root-cause diagnosis of BGP events. The two algorithms can be used together or separately. The only coupling between them is that Stemming can extract a subset of an event stream encompassing a routing incident, which can then be fed to TAMP to generate an animation to visualize that incident.

A. Visualization: TAMP Picture and Animation

TAMP stands for **T**hreshold **a**nd **M**erge **P**refixes. It can generate a picture or an animation of a set of BGP routes. Unlike projects that aim to map the whole Internet from a global perspective, TAMP's goal is to show inter-domain routing *as the routers see it*. A BGP router stores routes heard from its peers in a Routing Information Base (RIB). Each entry in the RIB refers to a single prefix, along with its path attributes such as BGP Nexthop and AS path to reach the prefix. Figure 1(a) illustrates RIBs for routers X and Y. Using the RIB entries, TAMP forms a virtual tree to represent the BGP routes known by a router at a particular time: the root is the router; the router is linked to each of the BGP Nexthops of the routes; the Nexthops are linked to the ASes they service; each AS is linked to the next downstream AS according to the AS paths; and the leaf ASes are linked to the prefixes they advertise. TAMP then assigns a weight to each edge of this virtual tree proportional to the number of unique prefixes carried on the edge. See Figure 1(b) for the

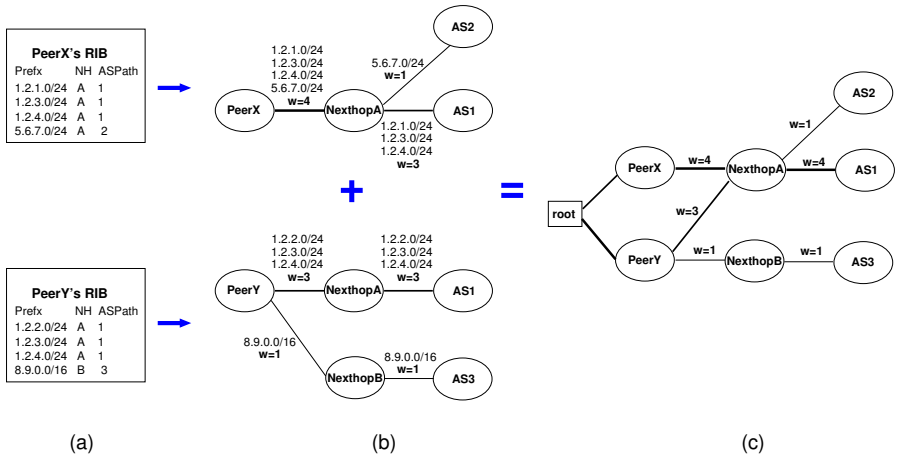


Fig. 1. Constructing a TAMP picture. Note the edge weight on NexthopA-AS1 of the combined tree in (c) is 4 not 6, because the edge carries 4 unique prefixes on it (1.2.1.0/24, 1.2.2.0/24, 1.2.3.0/24, 1.2.4.0/24).

individual TAMP trees for X and Y. TAMP constructs such a tree for each of a site's BGP edge routers, or core route reflectors for an ISP, and merges the trees into a graph. During this merging phase, TAMP does not simply add the edge weights together because a weight corresponds to the number of unique prefixes. It computes edge weights by combining common prefixes, i.e., it remembers the set of prefixes on each edge, performs a set union on the prefixes carried on the same edge from multiple trees, and uses the size of the union as the merged edge weight. Figure 1(c) is the combined TAMP graph for the routers X and Y. Data traffic would flow left-to-right. BGP information is flowing right to left. The thickness of an edge is proportional to how many prefixes are routed over that edge, not how much traffic is flowing over the edge. Note that TAMP is not limited to using all BGP routes at a router; the algorithm can map any set of routes, such as routes from a particular neighbor AS, or routes tagged with a specific LocalPref value, and so on. By appropriately choosing this set of routes, many problems can be diagnosed. We will show real-life examples in Section IV.

We use ATT's graphviz library [4] to layout a TAMP graph. For any realistic network, the TAMP graph described so far would be extremely bushy with most parts representing a negligible amount of prefixes, because the Internet topology has a well-connected core with high fan-outs toward the edges. Operationally, this picture would not be useful — even the most sophisticated layout algorithm would just generate a huge ink blob. The TAMP algorithm remedies this by pruning the graph so that only the most heavily used parts remain. Our default is to prune all edges and nodes which carry less than 5% of total prefixes in the graph. We found this default generates meaningful pictures for a range of networks, from universities to enterprises to ISPs. Figure 2 is a TAMP picture (with the default threshold) of Berkeley showing how it reaches the outside world using BGP. The leftmost rectangle is the TAMP root and represents the Berkeley campus (our

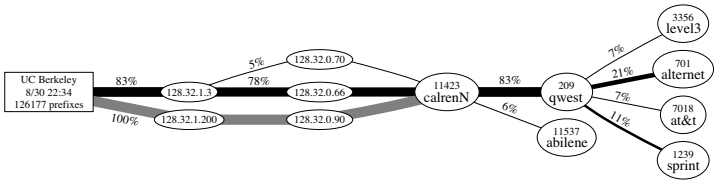


Fig. 2. TAMP picture of Berkeley's BGP.

REX recorder). 100% of prefixes comes from CalREN, 80% of that are from the commodity Internet through QWest, and then fanning out to other Tier-1 providers; 6% of the prefixes are from Abilene which is Internet2. TAMP is able to show high-level routing policies of a network. It would be difficult to see this information from routing table dumps without a TAMP picture.

TAMP also employs hierarchical pruning, in which increasing thresholds are applied as it walks down the graph farther away from the root. This is in response to feedback we received from network operators. An operator cares most and has control over elements in his own domain. All routers at his site are important to him, no matter how unsubstantial the number of prefixes carried by a router. Sometimes, a router announcing just a few prefixes can lead to a serious problem. Again, we will show real-life examples in Section IV.

A router's TAMP tree changes frequently and continuously over time as it receives BGP messages from its peers. Route announcement for a new prefix using an AS path that is not part of the TAMP tree creates additional branches to the tree. If the path is already part of the tree, the weight and width on each edge of the path need to be incremented. Likewise, route withdrawals can remove parts of the TAMP tree or decrease some edge parameters. Given a stream of BGP events, the TAMP algorithm tracks the routing changes expressed by the events to generate frames of TAMP pictures to form an animation. We do not attempt to display every single routing change in an animation; the human eye would not be able to see an edge losing 10,000 prefixes over a few seconds at granularity of one prefix at a time. Instead, the algorithm generates an animation with a fixed play duration of 30 seconds, regardless of the actual event timerange (which can range from seconds to days). The animation uses the standard 25 frames per second, each frame consolidating multiple routing changes on each edge of a TAMP graph.

TAMP animations use a number of additional visual cues. Figure 3 is a snapshot of a TAMP animation for a persistent MED route oscillation in ISP-Anon. At the bottom is an animation clock, displaying the time into the incident currently being shown. The plot to the right of the controls shows how the prefixes varied with time on whichever edge is selected in the TAMP graph. In this case, the selected edge is core1-b to 10.3.4.5, and since there is only one prefix, 4.5.0.0/16,

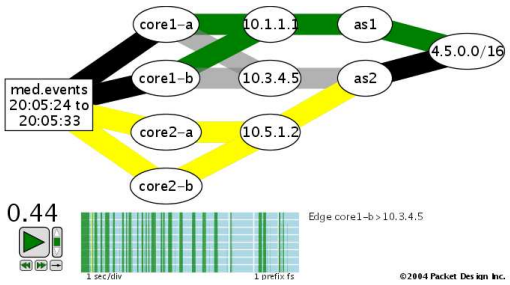


Fig. 3. TAMP animation snapshot of a persistent MED route oscillation in ISP-Anon.

the impulses on the plot tell us that the edge is flapping between carrying and not carrying the prefix. The edge colors indicate how the routes are changing: black means not changing; blue means the edge is losing prefixes; green means the edge is gaining prefixes; yellow means the prefix count is flapping too fast to animate; and an edge that has lost prefixes also has a gray shadow that indicates the largest number of prefixes it ever carried. The thickness of the non-gray part of an edge is proportional to the number of prefixes it is currently carrying.

B. Anomaly Detection: Stemming

BGP is extremely chatty because of its path vector nature. BGP sends out a million messages not because a million incidents happened. Instead, because the protocol can only talk at the prefix level, there is no way for it to say directly the one or two incidents that occurred, e.g. peering session reset, route flap, connectivity change. The challenge is to find the structure of the few incidents in the events. Stemming detects routing anomalies by finding the most strongly correlated components in a stream of BGP events, with each component representing a set of related routing changes. It is a statistical correlation algorithm modeled after principal component analysis.

A BGP event e is a route announcement (withdrawal) from a peer x for a prefix p with (old) path attributes Nexthop h and AS path $a_1 \dots a_n$. We can express e as a sequence as follows: $c = xha_1 \dots a_n p$. Let's call all such sequences comprising a BGP event stream $C = c_1, \dots c_m$. For each possible sub-sequence s of each c, the algorithm counts the number of times s appears in C. It then ranks all sub-sequences in descending order of their counts, and picks the highest ranking sub-sequence s'. The algorithm identifies the last pair of adjacent elements in this s' as the problem location. We term this pair of elements a *stem*. The set of prefixes P affected by this problem is the set of p in C containing s'. The set of events E which makes up this problem is the set of e containing any of the prefixes in P. In other words, E is a strongly correlated component in the event stream that represents a set of routing changes. We can apply the algorithm recursively to an event stream: the algorithm finds the first component E, removes E from the stream from consideration,

```
W 128.32.1.3   NEXT_HOP: 128.32.0.70 ASPATH: 11423 209 701 1299 5713    PREFIX: 192.96.10.0/24
W 128.32.1.3   NEXT_HOP: 128.32.0.66 ASPATH: 11423 11422 209 4519       PREFIX: 207.191.23.0/24
W 128.32.1.200 NEXT_HOP: 128.32.0.90 ASPATH: 11423 209 701 1299 5713    PREFIX: 192.96.10.0/24
W 128.32.1.200 NEXT_HOP: 128.32.0.90 ASPATH: 11423 209 1239 3228 21408  PREFIX: 212.22.132.0/23
W 128.32.1.3   NEXT_HOP: 128.32.0.66 ASPATH: 11423 209 701 705          PREFIX: 203.14.156.0/24
W 128.32.1.3   NEXT_HOP: 128.32.0.66 ASPATH: 11423 11422 209 1239 3602  PREFIX: 209.5.188.0/24
W 128.32.1.3   NEXT_HOP: 128.32.0.66 ASPATH: 11423 209 7018 13606       PREFIX: 12.2.41.0/24
W 128.32.1.3   NEXT_HOP: 128.32.0.66 ASPATH: 11423 209 7018 13606       PREFIX: 12.96.77.0/24
W 128.32.1.3   NEXT_HOP: 128.32.0.66 ASPATH: 11423 209 1239 5400 15410  PREFIX: 62.80.64.0/20
W 128.32.1.200 NEXT_HOP: 128.32.0.90 ASPATH: 11423 209 1239 5400 15410  PREFIX: 62.80.64.0/20
```

Fig. 4. Route withdrawals during an event spike.

then finds the next component, and so on.

Figure 4 is an illustration. There are 10 route withdrawals during an event spike of a million at Berkeley. Eight out of the 10 withdrawals share a common portion, 11423-209, while the remaining paths comprise different ASes. This pattern occurs because there was a failure that led to the withdrawals. The last edge of the common portion, in this case 11423-209, would be the failure location. If the failure was one hop down between 209 and 7018, the common portion would be 11423-209-7018, and the last edge, 209-7018, is the failure location.

One of Stemming's important characteristics is temporal independence: it does not try to infer causality in BGP events and does not depend on event ordering. Stemming is a correlation technique, and correlation is a well-defined property at any time-scale. For anomalies such as peering loss, session reset or leaked routes, the time-scale would be short, in the order of convergence times. So if Stemming looks at events encompassing tens of minutes, these anomalies, especially ones involving the Internet core, would show up as strongly correlated because of the massive route withdrawals and announcements. Some serious problems often do not result in sudden surges of BGP events though. For things like persistent route oscillation or flaky link, the continuous events might be mistaken as BGP noise. But if Stemming looks at events representing hours or days, these anomalies even involving just a single prefix would overwhelm other correlations and show up as the strongest one.

C. Performance

Table I shows sample running times of the TAMP and Stemming algorithms when applied on data from Berkeley and ISP-Anon. Our implementation is in C++ and executed on an Intel Pentium 4 machine with a 3.06 GHz CPU and 1.5GB of memory. For the TAMP picture column, running time denotes the time the TAMP algorithm took to compute a picture representing the listed number of routes and then pruned with the default threshold. Under TAMP animation, timerange is the actual time period encompassed by the listed number of events, i.e. the difference between the first and last event's timestamp. For Stemming, each event group is an actual event spike in the network associated with a real routing change

TAMP picture		TAMP animation			Stemming		
No. routes	Run time	No. events	Timerange	Run time	No. events	Timerange	Run time
230k	1.8 sec	1k	423 sec	0.5 sec	12k	189 sec	8.6 sec
115k	1.6 sec	10k	36 min	1.1 sec	57k	882 sec	9.5 sec
23k	0.5 sec	100k	7.6 hrs	9 sec	330k	16.3 min	17.3 sec
		1000k	33.6 hrs	78 sec			

(a) Berkeley

TAMP picture		TAMP animation			Stemming		
No. routes	Run time	No. events	Timerange	Run time	No. events	Timerange	Run time
1500k	7 sec	1k	226 sec	1.0 sec	214k	61.7 min	32.8 sec
750k	3.8 sec	10k	621 sec	1.6 sec	346k	51.7 min	34.1 sec
150k	1.5 sec	100k	2.3 hrs	9.4 sec	791k	1.7 hrs	35.2 sec
		1000k	20.5 hrs	88.5 sec			

(b) ISP-Anon

TABLE I
EXECUTION TIMES OF TAMP AND STEMMING ALGORITHMS.

or anomaly. Note that the timerange for similar number of events is much shorter in ISP-Anon than Berkeley, as BGP is a lot chattier in an ISP because of its peerings with other ISPs. Since Berkeley has fewer routes than ISP-Anon, its running times are shorter for the same number of events — the algorithm keeps track of significantly smaller BGP RIB and topology data structures for Berkeley. In these measurements, we ran the algorithms starting at the current state of the system. In other words, we do not include time to rebuild the data structures to move to any random point in time. This setup reflects how the algorithms would be used in a real-time system.

D. Additional Data Sources

A single type of data source is sometimes not enough information to decipher a complex network and its routing. In this section, we describe the integration of information from router configuration files, traffic flows and IGP routing data into our algorithms to combat the drawbacks with using BGP events alone in Internet routing anomaly diagnosis.

D.1 Integration with Internal Routing Policies

BGP routing policies local to an AS are defined through configuring routers, and are stored in configuration files at the routers. Policies affect routing decisions, but are not present in BGP events and are considered private to an AS. There are many types of routing policies. A router can define access lists to filter out routes from certain routers. An AS can configure its border routers to only advertise locally originated routes (e.g. routes with empty ASPATH attributes) to prevent becoming transit to other ASes. A router can choose to always prefer routes from a particular peer. The BGP

community attribute in particular is used heavily to dynamically influence routing decisions.

It is important to understand the interactions between intended policies and actual routing behaviors. Note that this is a step beyond automatically generating and validating configuration files for routers: even if each router is configured as intended, an unexpected change anywhere can put routing into an exceptional state, which when combined with policies can lead to suboptimal or even unacceptable behaviors. Section IV-D describes how while policies tied to the BGP community attribute is powerful when everything is functioning as anticipated, during instability the compounded interactions with the policies can lead to disastrous results.

We have deduced part of Berkeley's routing policies by studying its BGP events. However, without looking at the content of configuration files, it is difficult to know all of the complex policies defined internally in a network. We are working on automatically correlating routing changes with routing policies. One can imagine retrieving router configuration files from the two Berkeley BGP peers in question, and then correlate their policies with BGP events in real-time. What would their configuration files tell us? For router 128.32.1.3, the one on the rate-limiting path, it would have a command to assign a LOCALPREF value of 80 to ISP routes tagged with 11423:65350 from CalREN. On router 128.32.1.200, the one on the non-rate-limiting path, it would give a lower LOCALPREF of 70 to those ISP routes, and a default value of 100 to non ISP routes tagged with 11423:65300 (Internet2, CalREN members, etc). Recent work [5] presents a methodology for reverse engineering the design of a network through static analysis of router configuration files. We can use similar parsing techniques to extract out the above route preference policies. The results from Stemming can be matched with these policies. Stemming picks out the most strongly correlated component in the BGP events, which in this case, composes of route withdrawals tagged with 11423:65350 from 128.32.1.3 and announcements tagged with 11423:65300 from 128.32.1.200. Correlating the tags with the local preferences, we can pinpoint this costly policy interaction within Berkeley and give hints to network operations on how to handle the situation.

D.2 Integration with Traffic

Internet traffic display the "elephant and mice phenomenon" [6], in which a small percentage of prefixes, the elephants, accounts for majority of the traffic volume, and the rest of the prefixes, the mice, use only very little bandwidth. For example, 10% of the prefixes can be associated with 90% of the traffic, where 90% of the prefixes are tied to only 10% of the traffic. The algorithms in this paper use prefix counts and event counts as metrics, and weigh each prefix equally. Though useful on their own, the algorithms would provide additional capabilities if combined with traffic data as well.

In Section IV we discuss a "Load Balancing Unbalanced" incident at Berkeley: Berkeley's intention was to employ a simple scheme of splitting the prefix space evenly across two paths to load balance traffic onto the two rate limiters, but a misconfiguration resulted in one path carrying four times more prefixes than the other. However, because of the elephant and mice phenomenon, this incident can be more or less severe than the discrepancy in prefix counts depending on the traffic volume linked to the two sets of prefixes. Turning on Cisco Netflow on the outbound interfaces to the two rate limiters is one method to collect traffic flows information. This would tell us the unbalance in terms of bandwidth consumed on the links which is a more meaningful measure in Berkeley's case. However, to arrive at an effective load balancing situation would require a sequence trial-and-error steps. A common practice with a setup like Berkeley's would be to adjust the prefix splits, wait some time to see how the traffic flows on the two links change, and readjust until the desired effect is achieved. A better way is to correlate routing and traffic data and compute traffic volume for each routing prefix, and recompute the volume as routing and traffic changes. This would allow us to compute an more effective, fine-grained prefix load balancing without affecting the network with trial-and-error steps. We are in the process of integrating traffic data into our algorithms. In TAMP visualization, instead of weighing each prefix equally, edge weights would be computed based on traffic volume on the edges inferred by routing data combined with Netflow traffic flow records collected at selected border routers of a network.

There are also studies that show that BGP routing instability, or churn, does not affect much of the traffic. In AT&T's IP backbone, measurements during the month of March 2002 show that a list of popular websites which are responsible for most of the Internet traffic experienced mostly stable routing [7]. In Sprint's IP backbone, measurements during 2 days in August 2003 show that BGP churn had limited impact on outbound traffic, in particular, only 0.05% of the routing changes affect the elephant prefixes carrying 80% of the traffic [8]. Even if routing instability of elephant prefixes is rare, there is nothing that stops it from happening, and the effect is costly. For instance, a short-term route oscillation on a few elephant prefixes in an enterprise network can slosh most of the network's Internet traffic between exit border routers, thereby significantly degrading performance at the customers. We are currently enhancing the Stemming algorithm to do a weighted correlation on traffic volume tied to prefixes in a stream of events.

D.3 Integration with IGP

The BGP route selection process works with IGP to compute the best route for a prefix. It considers the reachability and IGP cost to the NEXTHOP attribute in a BGP route. A change in IGP such as link metric can cause a router to

reselect a different BGP best route. In this work, we feed only BGP events to the Stemming algorithm to discover an incident. We then use REX, which temporally synchronizes BGP and IGP routing messages from a network, to manually drill-down and determine whether IGP is part of the root-cause of an incident. The volume of IGP routing messages (e.g. LSAs in OSPF) is multiple orders of magnitude lower than BGP. This makes it convenient to correlate LSAs with a BGP incident after the incident is discovered. We are working on automating this process as part of Stemming.

IV. Results: Case Studies on Real Networks

We applied our routing anomaly detection and visualization algorithms in a variety of networks. It works well for both simple tree-like topology of an university or enterprise with relatively few BGP edge routers and a single provider, and also for rich, forest-like topology of a Tier-1 service provider with hundreds of core BGP routers peering with most of the other Tier-1s and many customers. In this section, we present results of the algorithms on Berkeley and ISP-Anon.

A. Load Balancing Unbalanced

Let us revisit Figure 2, a TAMP picture of Berkeley's BGP. Most of this picture is expected. However, when we showed this picture to Berkeley's network engineers, they found a previously undiagnosed misconfiguration. The BGP edge router 128.32.1.3 is configured to carry commodity Internet traffic, and for load balancing purposes, Berkeley simply splits the prefix address space in half onto two rate-limiting BGP Nexthops, 128.32.0.66 and 128.32.0.70. There was an error during this split, and the division turned out to be much skewed: 128.32.0.66 carried 78% of the total advertised prefixes, and 128.32.0.70 only 5%. It would be hard to detect this misconfiguration with a "show ip bgp" output at the router. "One picture says a million routes" is the power of TAMP.

B. Backdoor routes

Figure 5 is a TAMP picture of Berkeley drawn with hierarchical pruning: all BGP peers, Nexthops and neighbor ASes are shown, and the rest of the ASes are pruned with a 5% threshold. This picture exposes two backdoor routes between 128.32.1.222 and AT&T, via the Nexthop 169.229.0.157. Backdoor routes might have severe impact on a network. It can create a security breach unbeknownst to network administrators. Again, it would be easy to miss these two routes in a "show ip bgp" output.

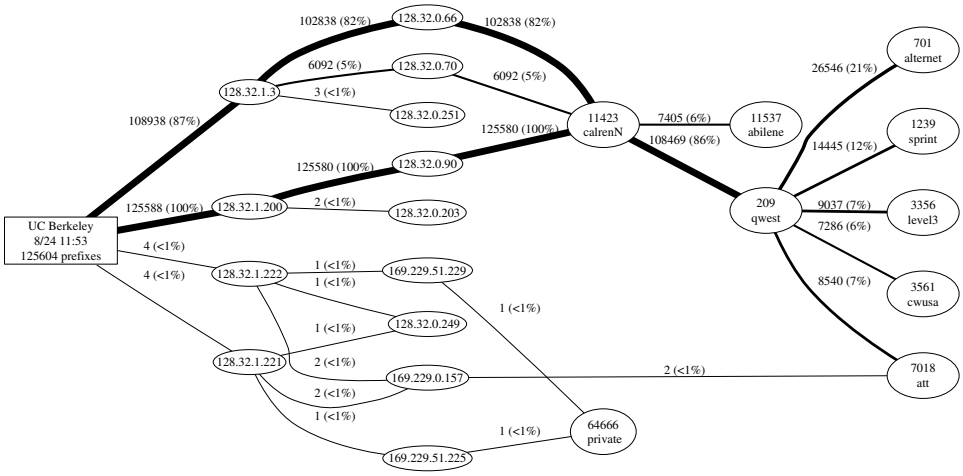


Fig. 5. TAMP visualization of Berkeley's BGP, showing 2 backdoor routes between 128.32.1.222 and AT&T.

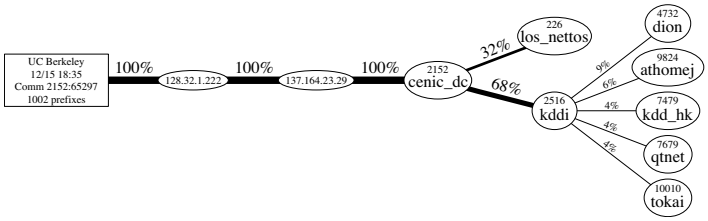
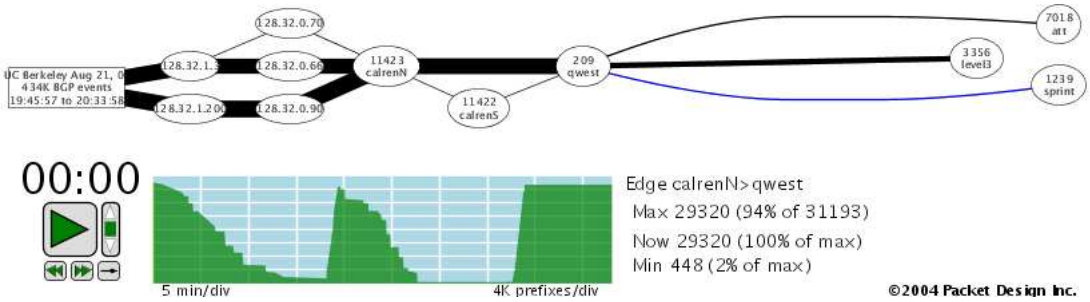


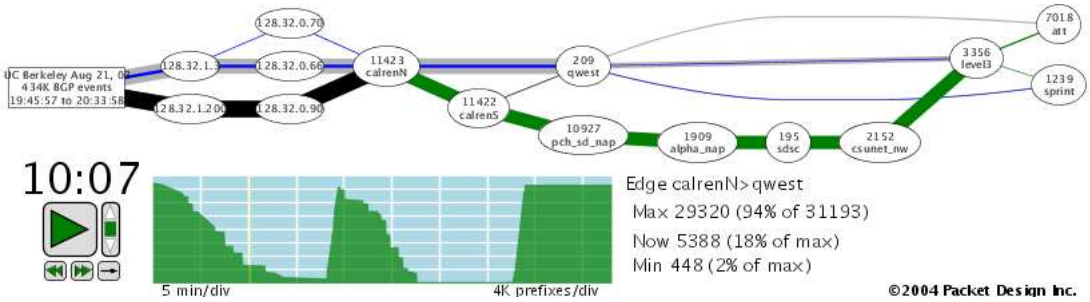
Fig. 6. Mis-tagging of community 2152:65297, which is supposed to be only attached to routes coming from Los Nettos peering via LAAP.

C. BGP Community Mis-tagging

TAMP shows the large-scale structure of any set of routes. One meaningful subset are routes tagged with a certain community. Many routing policies are tied to community values: route filtering, setting route preferences, route scoping, etc. It is useful to see if a community tag really means what documentation says it means. Figure 6 is a TAMP visualization of routes tagged with the CENIC community 2152:65297, heard at Berkeley in December 2003. CENIC attaches this value only to routes coming from the Los Nettos peering via LAAP [9], not to the ones received from customer links. We observe from Figure 6 that only 32% of the prefixes with 2152:65297 were from Los Nettos, where as 68% came from KDDI, a Japanese ISP. If this community is used to set higher priority to Los Nettos routes based on some agreement between Berkeley and Los Nettos, then KDDI would have also been given preferential treatment. We confirmed with CENIC that this is indeed an error. This mis-tagging was probably made during the consolidation of multiple AS numbers into just AS2152. CENIC has fixed the problem since then.



(a) Before leaked routes from CalREN's peers. All routes are from QWest through CalREN.



(b) During route leakages. The {128.32.1.3-128.32.0.66-11423-209} path has a gray shadow with a thin blue line which means it is losing prefixes. The {11423-11422-10927-1909-195-2152-3356} path is green which means it is gaining prefixes. The {128.32.1.200-128.32.0.90} path remains black means the number of prefixes it carries is not changing. The CalREN-QWest edge is selected in these snapshots.

Fig. 7. Leaked routes from CalREN's peers led to a 6 AS-hop path being preferred over the shorter CalREN-QWest path. Also, 128.32.1.3 stopped announcing the prefixes involved because of an interaction with BGP community filtering. All commodity Internet traffic would be routed through the non rate-limiting path.

D. Peer Leaking Routes

The first three incidents were manually detected by examining TAMP pictures. The remaining results were automatically detected by the Stemming algorithm illustrated with TAMP animations.

We observe a few incidents of export misconfiguration where leaked routes from CalREN's peers caused significant number of prefixes to move over to a much longer path and affected Berkeley. The leaked routes are preferred over the shorter paths because of CalREN's local preferences. Figure 7 contains snapshots of an animation showing one such incident (we encourage readers to visit [10] to play the actual animations referenced in this paper). This is a 500,000 event incident where 30,000 prefixes moved over, twice, from CalRENN-Qwest to Level-3 via a AS-hop path

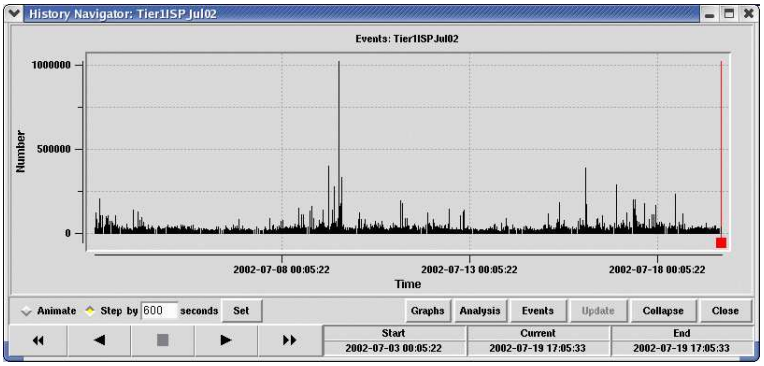
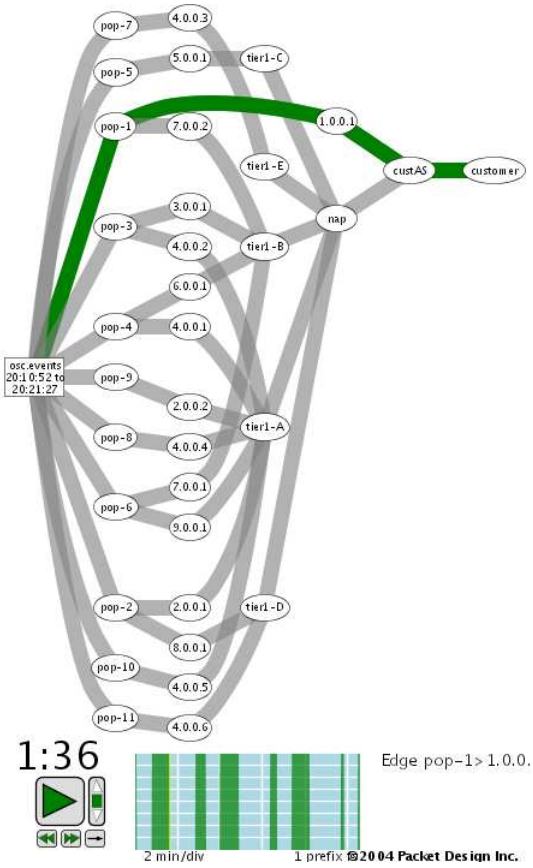


Fig. 8. BGP event rate at ISP-Anon.

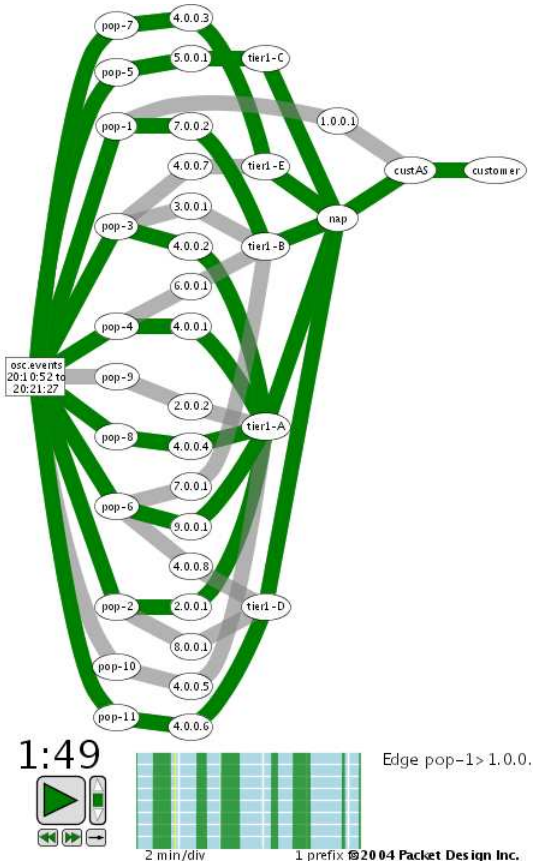
consisting of Packet Clearing House, Alpha NAP, San Diego Supercomputing Center and CENIC. We also observe another 220,000 BGP event incident where 30,000 prefixes moved over from CalRENN-Qwest to CalRENN's alternate ISP connectivity, Net Access via Global NAPs. Peer leaking routes can lead to suboptimal routing in the Internet. At Berkeley, the leaked routes, when interacted with a BGP community policy, led to some unexpected and costly consequences. From Figure 7(b), we see that whenever prefixes on the CalREN-QWest edge move over to another path, the BGP edge router 128.32.1.3 also stops announcing those prefixes. We know that an alternate path does exist because 128.32.1.200 advertises it. We confirmed with Berkeley that the intended behavior would be for 128.32.1.3 to advertise routes for the commodity Internet, thereby making the two rate-limiters 128.32.0.66 and 128.32.0.70 on the primary path to the outside world. However, since 128.32.1.3 is not announcing the alternate path, effectively all commodity Internet traffic ended up going through 128.32.0.90 which is not rate-limited. Berkeley does not rate-limit Internet2 traffic that goes toward Abilene. Thus 128.32.1.3, which uses the rate-limiter BGP Nexthops 128.32.0.66 and 128.32.0.70, filters out all routes except those for the commodity Internet. 128.32.1.3 uses BGP community from CalREN to distinguish between commodity Internet routes and other routes such as those for Internet2. When routes are not heard directly from QWest, CalREN does not attach the ISP's community on them. One can argue this is the correct behavior from CalREN's viewpoint, even though the routes do come from a commercial provider, Level3, at the end of the 6 AS-hop path. This type of interaction between routing anomalies and intended policy causing unexpected actual behavior is difficult if not impossible to diagnose without tools like Stemming and TAMP.

E. Continuous Customer Route Flapping

We now turn our focus to ISP-Anon. Figure 8 is a graph of BGP event rate at ISP-Anon during June–August 2002. The most serious problem is not in any of the event spikes in the graph. It shows up as low-grade BGP churn in the



(a) When direct path between ISP-Anon and customer is stable. The {pop1-1.0.0.1-custAS-customer} (green) is preferred. All other routes (gray) are suppressed.



(b) When the direct path fails. The BGP core routers all announce the 3 AS-hop paths (green) through another Tier-1 ISP and the NAP. For example, pop3 advertises the route {5.0.0.1-tier1C-nap-custAS-customer}.

Fig. 9. Continuous customer route flapping. The routing rapidly flaps between the state in (a) and (b). The animation itself shows interesting intermediate states of route preferences coming into effect. The timeline here only show a few minutes out of the 1.5 months of flapping.

"grass" of the graph. The problem is a persistent route oscillation between ISP-Anon and one of its customers. The event rate is too low for most tools to detect the problem, but Stemming had no trouble finding it and TAMP animation is able to clearly illustrate it. Figure 9 shows snapshots of the animation. This customer of ISP-Anon has a direct connection via next hop 1.0.0.1 but the associated BGP peering would not stay up – it was dropped and re-established every minute on the average. The customer also has a backup link via a NAP that is connected to all the other Tier-1 ISPs so when the one-hop direct path goes away things immediately fail over to a three-hop alternate via some other Tier-1. Since each

pop peers with different Tier-1s and each makes an independent decision, lots of different alternate paths are announced. The convergence details vary slightly event to event (depending on the relative timing of each core route reflector's updates from the access routers peering with the various downstream ISPs) but it takes about 20 seconds for everything to converge and generates about 200 BGP events per customer flap. This oscillation went on continuously for more than a month and a half.

F. Persistent Fast MED Oscillation

Let us revisit the TAMP animation in Figure 3. Stemming detected this MED oscillation with surprising intensity at ISP-Anon . There are four core route reflectors involved, two in each of two PoPs. Core1-a/b and Core2-a/b each have paths to 4.5.0.0/16 via AS2. Core1-a/b also have a path via AS1. ISP-Anon is accepting MEDs from AS2 and Core1 has the better MED. So Core1-a/b switch between the AS1 and AS2 paths as Core2-a/b announce/withdraw their AS2 route. In this case Core2-a/b are each announcing and withdrawing their AS2 route on the average of every 10 microseconds (100,000 times per second each — the links are colored yellow since the event rate is too fast to animate). This flood causes Core1-a and Core1-b to switch paths on the average every 10 milliseconds (100 times/second). The animation shows 10 seconds of this (note that the time scale on this animation is milliseconds while the others have been seconds or minutes). The actual event lasted for at least five days, continuously, and accounted for 95% of the ISP's BGP traffic. That is, this one prefix generated 20 times more IBGP traffic than all the rest of the Internet combined. This oscillation is the strongest correlation component detected by Stemming even when applied to a short timescale of a few minutes.

V. STATUS

We first presented this work at the North America Network Operators Group 30th meeting [11] and it was very well-received. The TAMP visualization technique and Stemming analysis algorithm have since been adopted by Packet Design's Route Explorer (REX) product as part of its BGP diagnostics and analytics solution. REX consolidates data from multiple routing protocols—currently it supports OSPF, ISIS, EIGRP, BGP and MPLS/VPNs—and computes a real-time, network-wide routing map. It allows an user to monitor the overall routing topology of a network as it changes, as well as providing a historical view. It also has a number of features for drilling down to show details of routing state. We have installed REX with our extensions at large universities, enterprises, public providers, and major Tier-1 ISPs. Preliminary reactions from these installation sites have been very positive. As they use TAMP and Stemming to detect

and diagnose BGP misbehavior, we hope to gather feedback from them and further improve our algorithms.

VI. Related Work

In this section, we discuss work most related to ours. The list here is by no means exhaustive.

To improve understanding of Internet routing dynamics, researchers at U.C. Davis's Elisha project [12] have developed a suite of visualization techniques, including modules to browse the timing and stability of BGP messages, and for observing router, link, and peer changes. The BGPlay Java plugin [13] from RIPE NCC visualizes BGP routing activity for a single prefix within a time interval. The TAMP techniques presented in this paper can animate any set of prefixes.

A number of researchers are investigating the problem of BGP root-cause analysis. Caesar et al [14] and Feldmann et al [15] both have developed algorithms that try to pinpoint the root causes of Internet routing dynamics. Using BGP data from multiple vantage points, their algorithms can distinguish among incidents that are correlated across time and prefixes. Teixeira and Rexford [16] and researchers at University of Minnesota [17] both point out the limitations of using BGP data from a single vantage point in identifying location and cause of routing changes, in which the results can be incomplete, and propose techniques to deal with them.

The above papers advocate global coordination among ASes, which would allow one AS to peek into another's network when diagnosing routing problems. Although a laudable goal, this inter-AS framework would require much commercial incentives to deploy since most ISPs guard their internal network data as secrets. Our work focuses on what is currently possible with commonly deployed interdomain routing practices within a network's scope of administrative control. Through real-life observations, we show that the operationally oriented information our algorithms provide is a useful first step in diagnosing routing problems in the Internet.

VII. Conclusions

We have presented challenges in understanding the inter-domain routing system of today's Internet. It is important to be able to diagnose routing problems, especially the subtle ones, quickly and accurately because they can cause harm to the Internet in many ways. This has motivated us to develop the visualization and analysis algorithms presented in this paper. Our results from applying the algorithms to both a Tier-1 ISP and a large university network show that the algorithms can help diagnose a wide range of routing anomalies. We also discovered unexpected interactions between intended routing policy and actual behavior through the algorithms, which would otherwise be next to impossible to

detect in an operational environment. The algorithms have been incorporated in a commercial product as part of a BGP diagnostics solution. There are installations of this products in a variety of network settings and have been well-received by the customers.

REFERENCES

[1] Yakov Rekhter, Tony Li, and Editors Susan Hares, "A border gateway protocol (bgp-4)," 10 2002, Internet Draft (work in progress). Available at http://www.ietf.org/internet-drafts/draft-ietf-idr-bgp4-22.txt.

[2] D. McPherson, V. Gill, D. Walton, and A. Retana, *Border Gateway Protocol (BGP) Persistent Route Oscillation Condition*, 2002, RFC-3345.

[3] "Packet Design, Inc.," http://www.packetdesign.com.

[4] "ATT's graphviz library," http://www.research.att.com/sw/tools/graphviz/.

[5] David Maltz, Geoff Xie, Jibin Zhan, Hui Zhang, Gisli Hjalmtysson, and Albert Greenberg, "Routing design in operational networks: A look from the inside," in *Proceedings of Sigcomm*, Portland, OR, September 2004.

[6] K. Papagiannaki, N. Taft, S. Bhattacharyya, P. Thiran, K. Salamatian, and C. Diot, " A Pragmatic Definition of Elephants in Internet Backbone Traffic," in *SIGCOMM Internet Measurement Workshop*, Marseilles, France, Nov. 2002.

[7] J. Rexford, J. Wang, Z. Xiao, and Zhang Y, "BGP Routing Stability of Popular Destinations," in *SIGCOMM Internet Measurement Workshop*, Marseilles, France, Nov. 2002.

[8] Sharad Agarwal, Chen-Nee Chuah, Supratik Bhattacharyya, and Christophe Diot, "The impact of BGP dynamics on intra-domain traffic," Sprint ATL Research Report RR03-ATL-111377, 2003.

[9] "CENIC network operations website: CENIC BGP communities," http://www.cenic.net/operations/documentation/BGPCommunities.shtml.

[10] "Packet Design, Inc.: Making Sense of BGP Animations," http://www.packetdesign.com/technology/presentations/nanog-30/index.htm.

[11] "The NANOG30 Meeting," February 2004, http://www.nanog.org/mtg-0402/index.html.

[12] Soon Tee Teoh, Kwan-Liu Ma, and S. Felix Wu, "A visual exploration process for the analysis of internet routing data," in *Proceedings of 14th IEEE Visualization Conference*, Seattle, WA, October 2003.

[13] "Bgplay – graphical visualisation of bgp updates," RIPE NCC.

[14] Matthew Caesar, L.Subramanian, and Randy H. Katz, "Root cause analysis of internet routing dynamics," Tech. Rep., U.C. Berkeley, November 2003.

[15] Anja Feldmann, Olaf Maennel, Z. Morley Mao, Arthur Berger, and Bruce Maggs, "Locating internet routing instabilities," in *Proceedings of Sigcomm*, Portland, OR, September 2004.

[16] Renata Teixeira and Jennifer Rexford, "A measurement framework for pinpointing routing changes," in *Proceedings of NetTs Workshop*, Portland, OR, September 2004.

[17] Jaideep Chandrashekar, Zhi-Li Zhang, and Haldane Peterson, "Fixing bgp, one as at a time," in *Proceedings of NetTs Workshop*, Portland, OR, September 2004.