Finding a Needle in a Haystack: Pinpointing Significant BGP Routing Changes in an IP Network

Jian Wu, Zhuoqing Morley Mao
University of Michigan

Jennifer Rexford
Princeton University

Jia Wang
AT&T Labs–Research

Abstract

The performance of a backbone network is vulnerable to *interdomain* routing changes that affect how traffic travels to destinations in other Autonomous Systems (ASes). Despite having poor visibility into these routing changes, operators often need to react quickly by tuning the network configuration to alleviate congestion or by notifying other ASes about serious reachability problems. Fortunately, operators can improve their visibility by monitoring the Border Gateway Protocol (BGP) decisions of the routers at the periphery of their AS. However, the volume of measurement data is very large and extracting the *important* information is challenging. In this paper, we present the design and evaluation of an online system that converts millions of BGP update messages a day into a few dozen *actionable* reports about significant routing disruptions. We apply our tool to two months of BGP and traffic data collected from a Tier-1 ISP backbone and discover several network problems previously unknown to the operators. Validation using other data sources confirms the accuracy of our algorithms and the tool's additional value in detecting routing disruptions.

1 Introduction

Ensuring good performance in an IP backbone network requires continuous monitoring to detect and diagnose problems, as well as quick responses from management systems and human operators to limit the effects on end users. Network operators need to know when destinations become unreachable to notify affected customers and track down the cause of the problem. When measurements indicate that links have become congested, operators may respond by modifying the routing protocol configurations to direct some traffic to other lightly-loaded paths. These kinds of measurements are also crucial for discovering weaknesses in existing network protocols, router implementations, and operational practices to drive improvements for the future. All of these tasks require effective ways to cull through large amounts of measurement data, often in real time, to produce concise, meaningful reports about changes in network conditions.

To track events inside their own network, operators collect measurements of data traffic, performance statistics, the internal topology, and equipment failures. The performance of a backbone network is especially vulnerable to *interdomain* routing changes that affect how data traffic travels to destinations in other Autonomous Systems (ASes). For example, a link failure in a remote AS could trigger a shift in how traffic travels through a network, perhaps causing congestion on one or more links. Fortunately, operators can gain additional visibility into the interdomain routing changes by monitoring the Border Gateway Protocol (BGP) decisions of routers at the periphery of their AS. In this paper, we address the challenge of analyzing a large volume of BGP update messages from multiple routers in real time to produce a small number of meaningful alerts for the operators.

In addition to the large volume of data, producing useful reports is challenging because: (i) BGP update messages show the changes in AS-level paths without indicating why or where they originated, (ii) a single network event (such as a failure) can lead to multiple update messages during routing protocol convergence, (iii) a single network event may affect routing decisions at multiple border routers, and (iv) a single event may affect multiple destination prefixes. Having a small number of reports that highlight only *important* routing changes is crucial to avoid overwhelming the operators with too much information. The reports should focus on routing changes that disrupt reachability, generate a large number of update messages, affect a large volume of traffic, or are long-lived enough to warrant corrective action. These concerns drive the design of our system. We have evaluated our system on two months of data from a tier-1 ISP and discovered several important problems that were previously unknown. Our system analyzes millions of

BGP update messages per day to produce a few dozen actionable reports for the network operators.

Despite some high-level similarities, our approach differs markedly from recent work on root-cause analysis of BGP routing changes [6, 8, 13, 15, 30]. These studies analyze streams of BGP update messages from vantage points throughout the Internet, with the goal of inferring the location and cause of routing changes. Instead, we consider BGP routing changes seen *inside* a single AS to identify—and quantify—the *effects* on that network. Realizing that root-cause analysis of routing changes is intrinsically difficult [27], we search only for explanations of events that occur close to the AS—such as internal routing changes and the failure of BGP sessions with neighboring domains—and mainly focus on alerting operators to the performance problems they can address. Hence, our approach is complementary to previous work on root-cause analysis, while producing results of direct and immediate use to network operators.

In the next section, we present background material on BGP, followed by an overview of our system in Section 3. In Section 4, we group BGP update messages into routing *events*. We identify persistently flapping prefixes and pinpoint the causes. In Section 5, we introduce the concept of a *route vector* that captures the best BGP route for each prefix at each border router. We identify five types of routing changes that vary in their impact on the traffic flow. In Section 6 we group events by type to identify frequently flapping prefixes, BGP session resets, and internal routing disruptions; we validate our results using RouteViews data, syslog reports, and intradomain topology data. In Section 7, we use prefix-level traffic measurements to estimate the impact of the routing changes. Section 8 shows that our system operates quickly enough to generate reports in real time. Section 9 presents related work, and Section 10 concludes the paper.

2 BGP Overview

The Border Gateway Protocol (BGP) [21] is the routing protocol that ASes use to exchange information about how to reach destination address blocks (or *prefixes*). Three key aspects of BGP are important for our study:

Path-vector protocol: Each BGP advertisement includes the list of ASes along the path, along with other attributes such as the next-hop IP address. By representing the path at the AS level, BGP hides the details of the topology and routing inside each network.

Incremental protocol: A router sends an advertisement of a new route for a prefix or a withdrawal when the route is no longer available. Every BGP update message is indicative of a routing change, such as the old route disappearing or the new route becoming available.

1. Ignore if the next hop is unreachable;
2. Highest local preference;
3. Shortest AS path;
4. Lowest origin type;
5. Lowest Multiple-Exit-Discriminator (MED) value among routes from same AS;
6. eBGP routes over iBGP routes;
7. Lowest IGP cost (“hot-potato”);
8. Lowest router ID;

Table 1: BGP decision process

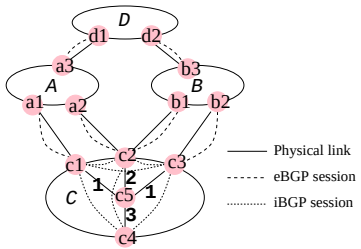

Figure 1: Interaction of routing protocols in AS C

Policy-oriented protocol: Routers can apply complex policies to influence the selection of the best route for each prefix and to decide whether to propagate this route to neighbors. Knowing why a routing change occurs requires understanding how policy affected the decision.

To select a single best route for each prefix, a router applies the decision process [21] in Table 1 to compare the routes learned from BGP neighbors. In backbone networks, the selection of BGP routes depends on the interaction between three routing protocols:

External BGP (eBGP): The border routers at the periphery of the network learn how to reach external destinations through eBGP sessions with routers in other ASes. A large network often has multiple eBGP sessions with another AS at different routers. This is a common requirement for two ASes to have a peering relationship, and even some customers connect in multiple locations for enhanced reliability. For example, Figure 1 shows AS C has two eBGP sessions with AS A and two eBGP sessions with AS B. As a result, there are three egress points to destinations in AS D.

Internal BGP (iBGP): After applying local policies to the eBGP-learned routes, a border router selects a single best route and uses iBGP to advertise the route to the rest of the AS. In the simplest case, each router has an iBGP session with every other router (*i.e.,* a *full-mesh* iBGP configuration). In Figure 1, the router $c4$ learns a two-hop AS path to destinations in AS D from three routers $c1$, $c2$, and $c3$.

Interior Gateway Protocol (IGP): The routers inside the AS run IGP to learn how to reach each other. The two most common IGPs are OSPF and IS-IS, which compute shortest paths based on configurable link weights. The

routers use the IGP path costs in the seventh step in Table 1 to select the *closest* egress point. In Figure 1, the number near each link inside AS C indicates the IGP cost of the link. Based on the decision rules, $c4$ prefers the routes through $c1$ and $c3$ over the route through $c2$ due to the smaller IGP path costs.[1]

The decision process in Table 1 allows us to compare two routes based on their attributes. We exploit this observation to determine whether a router switched from a better route to a worse route, or vice versa.

3 System Architecture

In this section, we describe how to track the BGP routing changes in an AS. Then, we present an overview of our system and describe the data we collected from a Tier-1 ISP backbone to demonstrate the utility of our tool.

3.1 Measurement Infrastructure

The routers at the edge of an AS learn BGP routes via eBGP sessions with neighboring domains, and then send update messages to other routers inside the AS via iBGP sessions. These border routers have complete visibility into external and internal routing changes. Ideally, each border router would provide a complete, up-to-date view of *all* routes learned from eBGP and iBGP neighbors. This data would allow our system to emulate the BGP decision process of each router, to understand why a router switched from one BGP route to another. Unfortunately, acquiring a timely feed of all eBGP updates received from neighboring ASes is difficult in practice.[2]

In this study, we analyze routing changes using only the data readily available in today's networks—a feed of the *best* route for each prefix from each border router. Our monitor has an iBGP session with each border router to track changes to the best route over time. A daily snapshot of the routing table from each border router is also collected to learn the initial best route for each prefix.

Since routing changes can have a significant effect on the distribution of the traffic over the network, traffic measurements are very useful for quantifying the *impact* of a routing change. In our measurement infrastructure, the monitor receives a feed of prefix-level traffic statistics from each border router. Because our analysis focuses on how routing changes affect the way traffic *leaves* the network, we collect the outgoing traffic on the edge links emanating from the border routers.

3.2 System Components

Our troubleshooting system analyzes BGP routing changes visible from inside a single AS and quantifies the effects on the network. The system is designed to operate *online* so operators may take corrective actions to improve network performance. For ease of presentation, we describe the functionality of our system in four distinct stages, as illustrated in Figure 2:

RouteTracker (Section 4): The first module merges the streams of BGP updates from the border routers and identifies routing *events*—groups of update messages for the same prefix that occur close in time. Along the way, the module identifies prefixes that flap continuously.

EventClassifier (Section 5): The second module classifies the routing events in terms of the kind of routing change and the resulting impact on the flow of traffic through the network. For example, we define a category called *internal disruption* that pinpoints the events caused by internal topology changes.

EventCorrelator (Section 6): The third module identifies related events by clustering over time and prefixes. In contrast to previous studies [6, 8, 13, 15, 30], we focus mainly on events that occur very close to the network (*e.g.,* eBGP session resets or internal disruptions) and have a significant impact on traffic. In addition, our correlation algorithms consider whether the border routers switched from a better route to a worse route, or vice versa—information not readily available in eBGP data feeds used in previous work on BGP root-cause analysis.

TrafficMeter (Section 7): The last module estimates the impact of routing changes on the flow of traffic, to draw the operators' attention to the most significant traffic shifts. Using prefix-level measurements of the traffic leaving the network, TrafficMeter computes a traffic weight that estimates the relative popularity of each prefix. The module predicts the severity of each event cluster by adding the weights of the affected prefixes.

In moving from raw updates to concise reports, we apply time windows to combine related updates and events, and thresholds to flag clusters with significant traffic volumes. We use our measurement data and an understanding of BGP dynamics to identify appropriate time windows; the threshold values reflect a trade-off between the number and significance of the disruptions we report.

3.3 Applying the System in a Tier-1 ISP

We have applied our prototype to a Tier-1 ISP backbone with hundreds of border routers connecting to customer and peer networks. Although we would ideally have iBGP sessions with all border routers, we could only collect data from the routers connecting to peer networks. Still, the BGP routing changes at these routers give us a unique view into the effects of BGP routing changes in the larger Internet on the ISP network. In addition, these border routers receive reachability information about customer prefixes via iBGP sessions with other routers, allowing us to analyze changes in how

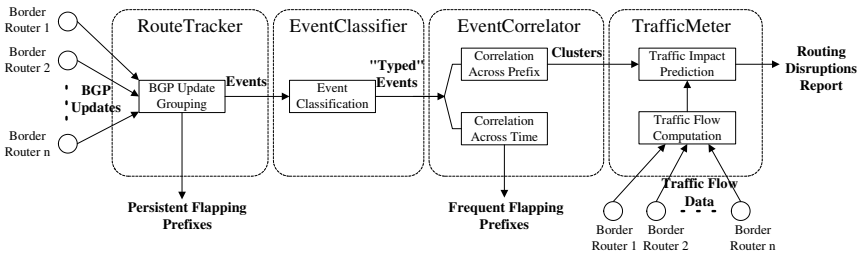

Figure 2: System design

Component	Reduction	Factor
RouteTracker	updates → events	15.2
EventCorrelator	events → clusters	31.7
TrafficMeter	clusters → "important" clusters	327.6
Total	updates → "important" clusters	158460

Table 2: Incremental information reduction

these border routers would direct traffic via customers. On a few occasions, our monitor experienced a temporary disruption in its iBGP connectivity to a border router; we preprocessed the BGP feeds as suggested in [22, 29] to remove the effects of these session resets.

The traffic data is collected from every border router by enabling Cisco's Sampled Netflow [1] feature on all links. To reduce the processing overhead, flow records are sampled using techniques in [19]. Although sampling introduces inaccuracies in measuring small traffic volumes, this does not affect our system since we only use the traffic data to identify large traffic disruptions.

As shown in Table 2, our system significantly reduces the volume of data and produces only a few dozen large routing disruptions from millions of BGP updates per day from the periphery of the network. "Important" clusters in the table are clusters that affect more than 1% of total traffic volume in the network. In the remainder of the paper, we present detailed results from the routing and traffic data collected continuously from August 16, 2004 to October 10, 2004—an eight-week period.

4 Tracking Routing Changes

In this section, we describe how we transform raw BGP update messages into routing events. We merge streams of updates from many border routers and identify changes from one stable route to another by grouping update messages that occur close in time. Along the way, we generate a report of prefixes that flap continuously.

4.1 Grouping BGP Updates into Events

A single network disruption, such as a link failure or policy change, can trigger multiple BGP messages as part of

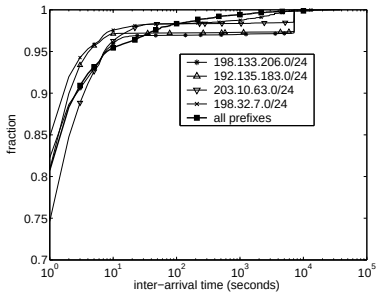

Figure 3: CDF of the BGP update inter-arrival time

the convergence process [3, 4]. The intermediate routes are short-lived and somewhat arbitrary, since they depend on subtle timing details that drive how the routers explore alternate paths. To generate reports for the operators, we are interested in the change from one stable route to another rather than the details of the transition. As such, we group BGP updates for the same prefix that occur close together in time. Although previous studies, in particular BGP root-cause analysis, have followed a similar approach [6, 8, 13, 20, 22], we group the updates across *all* of the border routers since a single network disruption may cause multiple border routers to switch to new routes, and we wish to treat these as a single event.

We define an *event* as a sequence of BGP updates for the same prefix from any border router where the inter-arrival time is less than a predefined *event timeout*. Careful selection of the event-timeout value is important to avoid mistakenly combining unrelated routing changes or splitting a single change into two events. An appropriate event-timeout value can be determined by characterizing the inter-arrival time of BGP updates in the network. For a controlled experiment, we analyze the inter-arrival times of BGP updates for public *beacon* prefixes that are advertised and withdrawn every two hours [17]; we also study the dynamics of the entire set of prefixes.

Figure 3 presents the cumulative distribution of the inter-arrival time of BGP updates for four beacons received from all of the border routers during a three-week period starting August 16, 2004, with the x-axis plot-

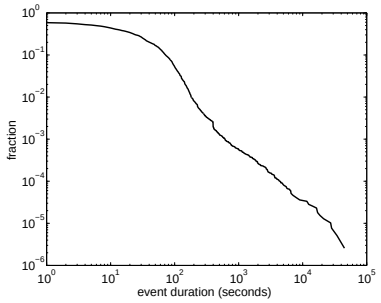

Figure 4: CCDF of event duration on a log/log scale

ted on a logarithmic scale. More than 95% of the inter-arrival times are within a few tens of seconds; then the curves flatten until the inter-arrival time is around 7,000 seconds reflecting the two-hour advertisement period. In addition, previous studies have shown that the path-exploration process is often regulated by a 30-second *MinRouteAdvertisementInterval* (MRAI) timer [14]. As such, we choose an event timeout of 70 seconds, allowing the difference between the arrival times of updates at different vantage points to be as large as two MRAIs plus a small amount of variance. Looking across all prefixes in our dataset, about 98% of the updates arrive less than 70 seconds after the previous update.

4.2 Detecting Persistent Flapping

Certain prefixes never converge to a stable path due to persistent routing instabilities. Persistent flapping disrupts the reachability of the destination and imposes a significant BGP processing load on the routers, making it important for operators to detect and fix these problems. However, if we group updates for a flapping prefix using a 70-second timeout, the grouping process would continue indefinitely. Instead, we generate a report once a sequence of updates exceeds a *maximum* duration, defined as the *convergence timeout*.

The convergence-timeout value should be large enough to account for reasonable convergence delays and yet small enough to report persistent flapping to the operators in a timely fashion. To identify an appropriate value, Figure 4 plots the complementary cumulative distribution function (CCDF) of event duration for the BGP updates in our network, with both axes on a logarithmic scale. More than 99% of events last less than a few hundred seconds, consistent with the findings in [3] that BGP typically takes less than three minutes to converge. As such, we select a convergence-timeout value of 600 seconds (10 minutes) for reporting flapping prefixes.

By applying our RouteTracker module to eight weeks of measurement data, we generated reports for about 23 prefixes per day, on average, though the number was as

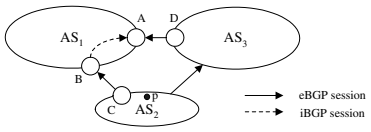

Figure 5: Persistent flapping due to failure of link B–C

low as 7 on one day and as high as 46 on others. These persistently flapping prefixes were responsible for 15.2% of the total number of BGP update messages over the two-month period, though the proportion varied significantly from day to day (from 3.2% to 44.7%). These results were especially surprising given that all of the border routers were running route-flap damping [5], which is meant to suppress repeated updates of the same prefix. We identified three main causes of persistent flapping:

Unstable interface/session: Using syslog data [16] from the border routers, we determined that 3% of these updates (0.456% of the total number of updates) were caused by repeated failures of a flaky edge link or eBGP session. The prefixes were advertised each time the link/session came online, and withdrawn when the link/session failed. In Figure 5, the routers in AS_1 prefer the BGP route advertised by the customer AS_2 over the BGP route advertised by the peer AS_3. However, a flaky link between routers B and C would lead the routers in AS_1 to repeatedly switch between the stable route via AS_3 and the unstable route via AS_2. Route-flap damping did not stop AS_1 from using the unstable route from AS_2 for two reasons: (i) today's routers reinitialize the damping statistics associated with an eBGP session after a session reset and (ii) routers do not perform route-flap damping on iBGP sessions. In the short term, operators could respond to these cases by disabling (and ultimately repairing) the flaky link or session; in the longer term, router vendors could change the implementation of route-flap damping to prevent the persistent flapping.

MED oscillation: Through closer inspection of the BGP update messages and discussions with the operators, we determined that 18.3% of these updates (2.78% of the total) were caused by protocol oscillation due to the Multiple Exit Discriminator attribute. Unlike the other steps in the decision process in Table 1, the MED comparison is applied only to routes with the same next-hop AS. As a result, the BGP decision process does *not* impose an ordering on the routes in the system: a router may prefer route a over route b, b over c, and c over a. In the absence of an ordering of the routes, the routers may switch continuously between routes [18, 25]. Upon detecting a MED oscillation problem, the operators can request that the neighboring AS use a different mechanism to express its preferences for where it wants to receive the traffic destined for these prefixes (*e.g.*, RFC 1998 [10]).

Conservative flap-damping parameters: The remaining 78.6% of these updates (11.9% of the total) correspond to repeated advertisements and withdrawals by a neighboring AS. By inspecting the configuration of the routers, we verified that the flap-damping parameters assigned for these prefixes were not sufficient to dampen the instability. Using different parameters for different prefixes is not uncommon and is, in fact, recommended [2]. For example, ASes are advised to more heavily penalize the (many) smaller address blocks and to disable damping on critical prefixes (*e.g.,* the subnets that contain the Internet's root DNS servers). Upon noticing persistent flapping that is evading the damping algorithm, the operator could contact the neighboring AS to investigate the root cause or tune the router configuration to apply more aggressive damping parameters.

5 Classifying Routing Changes

In this section, we describe how we classify events to generate useful reports for the operators and to facilitate the clustering of related events in the next section. Since the current measurement infrastructure collects the BGP data only from the border routers connecting to peer networks, the following analysis is applied to the prefixes learned exclusively from peer ASes.

5.1 Merging Routes from Border Routers

To handle the large volume of BGP data arriving from the many border routers, EventClassifier needs a succinct representation of the routing state as it evolves over time. Rather than considering every BGP attribute, we focus our attention on how traffic entering at a border router would leave the AS en route to the destination prefix p. A border router BR_j may select a route R_p^j learned directly from one of its eBGP neighbors; in this case, we say that BR_j has route R_p^j with the next-hop address $nhop_p^j$ corresponding to the eBGP neighbor and a $flag_p^j$ of **e** for external. Alternatively, a border router BR_j may select as R_p^j a route learned via iBGP from another border router, resulting in a next-hop address $nhop_p^j$ of the remote border router and a $flag_p^j$ of **i** for internal. In a network with n border routers $BR_1, BR_2, \ldots, BR_n$, we have a route vector (r-vector) for prefix p of

$$RV_p = \langle R_p^1, R_p^2, ..., R_p^n \rangle$$

where the jth element $R_p^j = (nhop_p^j, flag_p^j)$ represents the *best* route for prefix p at router BR_j. By analyzing the evolution of RV_p, we can identify and classify the routing changes that affect how traffic leaves the AS, while ignoring changes in other BGP attributes (*e.g.,* downstream AS path or BGP community) that are beyond the operators' control.

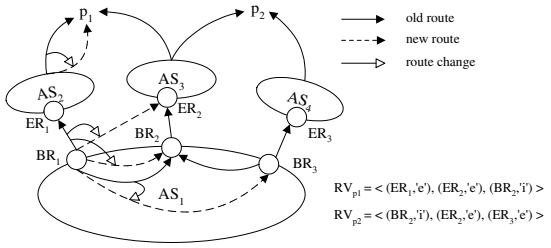

Figure 6: R-vector element changes

5.2 Classifying Routing Events

When the network changes from one set of stable routes to another, comparing the old and new r-vectors (RV_p^{old} and RV_p^{new}, respectively) sheds light on the reason for the change and the effects on the traffic. We first describe the types of changes that each border router might experience and then present five event categories that consider the behavior across all of the routers.

5.2.1 Types of Events at One Border Router

To illustrate the types of routing events, Figure 6 shows examples for two destination prefixes. For prefix p_1, border routers BR_1 and BR_2 have eBGP-learned routes through AS_2 and AS_3, respectively; border router BR_3 selects an iBGP-learned route through BR_2. For prefix p_2, border routers BR_2 and BR_3 have eBGP-learned routes through AS_3 and AS_4, respectively; border router BR_1 selects an iBGP-learned route through BR_2. The dashed lines represent different ways an event can affect BR_1's routing decision, as summarized in Table 3:

No change: The border router BR_j may undergo a transient routing change only to return to the same stable best route. More generally, the BGP route may change in some attribute that is not captured in R_p^j. In Figure 6, a change in how AS_2 reaches p_1 does not necessarily change BR_1's decision to direct traffic via AS_2. For all of these scenarios, traffic entering the network at router j destined for the prefix p would continue to flow through the AS in the same way.

Internal path change: An internal event may cause a router to switch from one egress point to another. In this case, router j uses an iBGP-learned route before and after the routing change (*i.e.,* $flag_p^{j,new} = flag_p^{j,old} =$**i**) but with a different next-hop router (*i.e.,* $nhop_p^{j,new} \neq nhop^{j,old}$). In Figure 6, a change in the IGP topology could make BR_1 see BR_3 as the *closest* egress point for reaching prefix p_2, instead of BR_2.

Loss of egress point: An external event may cause a route to disappear, or be replaced with a less attractive alternative, forcing a border router to select an iBGP route. In this case, a router BR_j has $flag_p^{j,old} =$**e** and

Type of Change for R_p^j	Definition
No change	$flag_p^{j,old} = flag_p^{j,new}$ $nhop_p^{j,old} = nhop_p^{j,new}$
Internal path change	$flag_p^{j,old} = flag_p^{j,new} =$**i**, $nhop_p^{j,old} \neq nhop_p^{j,new}$
Loss of egress point	$flag_p^{j,old} =$**e**, $flag_p^{j,new} =$**i**
Gain of egress point	$flag_p^{j,old} =$**i**, $flag_p^{j,new} =$**e**
External path change	$flag_p^{j,old} = flag_p^{j,new} =$**e**, $nhop_p^{j,old} \neq nhop_p^{j,new}$

Table 3: The types of change for r-vector element R_p^j

Event Category	Events	Updates	Upd./Ev.
Distant/transient disruption	50.3%	48.6%	12.6
Internal disruption	15.6%	3.4%	2.9
Single external disruption	20.7%	7.9%	5.0
Multiple external disruption	7.4%	18.2%	32.0
Loss/gain of reachability	6.0%	21.9%	47.9

Table 4: Event distribution in updates

$flag_p^{j,new} =$**i**. In Figure 6, suppose AS_2 withdraws its route for p_1 and that BR_1 has no other eBGP-learned routes; then, BR_1 would select the iBGP-learned route from BR_2. This routing change would force the traffic that used to leave the network at BR_1 to shift to BR_2.

Gain of egress point: An external event may cause an eBGP-learned route to appear, or be replaced with an attractive alternative, leading a border router to switch from an iBGP-learned route to an eBGP-learned one. In this case, a router BR_j has $flag_p^{i,old} =$**i** and $flag_p^{j,new} =$**e**. In Figure 6, suppose AS_2 starts advertising a route to p_1 again; then, BR_1 would start using the eBGP-learned route, causing a shift back to BR_1.

External path change: An external event may cause a router to switch between eBGP-learned routes with different next-hop ASes. In this case, the $flag_p^j$ remains at **e** while the next hop changes (*i.e.*, $nhop_p^{j,new} \neq nhop_p^{j,old}$). In Figure 6, suppose AS_2 withdraws the route for p_1, causing BR_1 to switch to an eBGP-learned route from AS_3. Then, BR_1 would start directing traffic to a different egress link at the same router.

5.2.2 Classes of Route-Vector Changes

Since each of the n elements in the r-vector can have five different types of changes, routing events could fall into 5^n different categories, which would be extremely unwieldy for generating reports for network operators. Instead, we classify the events based on the *severity* of the impact on the traffic, leading to five disjoint categories:

Distant/transient disruption: Some events do not have any influence on the flow of traffic through the AS. We define an event as *distant or transient disruption if each element of the r-vector has "no change."* A distant routing change that occurs more than one AS hop away does not affect the R_p^j values. A transient disruption may cause temporary routing changes before the border routers converge back to the original BGP routes. These events are worthwhile to report because the downstream routing change may affect the end-to-end performance (*e.g.*, by changing the round-trip time for TCP connections) and the convergence process may lead to transient performance problems that can be traced to the routing event. As shown in Table 4, this category explains about half of the events and half of the BGP update messages; these events trigger an average of 12 or 13 update messages for the BGP convergence process.

Internal disruption: An internal event can cause a router to switch from one internally-learned route to another. We define an event as an *internal disruption if the change of each of the elements in its r-vector is either of type "no change" or of type "internal path change", with at least one element undergoing an "internal path change."* Caused by a change in the IGP topology or an iBGP session failure, these events are important because they may cause a large shift in traffic as routers switch from one egress point to another [27, 28]. As shown in Table 4, internal disruptions account for about 15% of the events and just 3.4% of the updates; on average, an internal event triggers just a few iBGP update messages as some routers switch from one existing route to another.

Single external disruption: Some events affect the routing decision at a single border router for an eBGP-learned route. We define an event as a *single external disruption if only one r-vector element has a change of type "loss of egress point," "gain of egress point," or "external path change."* Typically, an ISP has eBGP sessions with a neighboring AS at multiple geographic locations, making it interesting to highlight routing changes that affect just one of these peering points. These kinds of events cause a shift in traffic because routers are forced to select an egress point that is further away [12]. For example, a single external disruption may arise because an eBGP session between the two ASes fails, forcing the border router to switch to a less-attractive route. As shown in Table 4, these disruptions account for over 20% of the events and nearly 8% of the updates; since these localized events affect a single router, the number of update messages per event is limited.

Multiple external disruptions: In contrast to the previous category, some events affect more than one border router. We define an event as a *multiple external disruption if multiple r-vector elements have a change of type "loss of egress point," "gain of egress point," or "external path change," and the r-vector includes at least one eBGP-learned route before and after the event.*[3] In Figure 6, if the owners of prefix p_1 changed providers to start using AS_4 instead of AS_2 and AS_3, every border routers

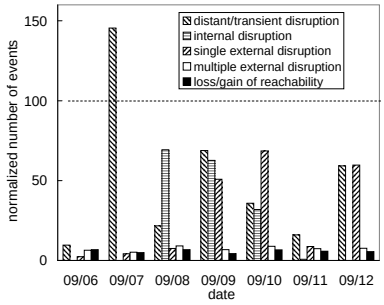

Figure 7: The (normalized) # of daily events by category.

in AS_1 would experience a disruption. As shown in Table 4, this category accounts for just over 7% of events and 18% of updates; the large number of update messages stems from the convergence process where multiple border routers must explore alternate routes.

Loss/gain of reachability: An event may cause a prefix to disappear, or become newly available. We define an event as *loss of reachability if every r-vector element with an external route experiences a "loss of egress point."* A loss of reachability is extremely important because it may signify a complete loss of connectivity to the destination addresses, especially if the routers have no route for other prefixes (*e.g.*, supernets) covering the addresses. Similarly, we define an event as *gain of reachability if initially no eBGP-learned routes exist and at least one r-vector element experiences a "gain of egress point."* In some cases, the *gain* of reachability is indicative of a problem, if the network does not normally have routes for that prefix. For example, a neighboring AS may mistakenly start advertising a large number of small subnets; overloading the memory resources on the router may have dire consequences, such as crashing the network [7]. As shown in Table 4, this category accounts for 6% of the events and nearly 22% of the update messages; the gain or loss of reachability often triggers a large number of update messages as every border router explores the many alternate routes.

Overall, the severity of the external events increases from single external disruptions to multiple external disruption, and ultimately to loss/gain of reachability. In general, the number of events in the "loss/gain of reachability" and "multiple external disruption" is stable over time, whereas the other categories vary significantly. Figure 7 shows the number of daily events (where 100 represents the average number of events per day over the eight-week study) for each event category during the week of September 6-12, 2004. For example, September 7 had a large number of distant/transient disruptions, and some days see a much larger number of internal disruptions and single external disruptions than others. The high variability arises from the fact that network disruptions can occur at arbitrary times and may affect a large number of destination prefixes, as discussed in the next section. Given the high variability in the number and type of events, predicting them in advance and overprovisioning for them is very difficult, making it even more important for operators to learn about disruptions as they occur to adapt the configuration of the network.

6 Grouping Related Events

In this section, we describe how to identify related events across *time* and *prefixes*. By clustering events across time for the same prefix, we identify destination prefixes that have unstable routes. By clustering events of the same type across prefixes, we group events that appear to have a common cause. We present techniques to identify groups of prefixes affected by hot-potato routing changes and eBGP session resets, which are responsible for many of the large clusters. We validate our inferences using RouteViews data [24], syslog reports [16], and an independent analysis [28] of internal topology changes.

6.1 Frequently Flapping Prefixes

Some destination prefixes undergo frequent routing changes that introduce a large number of events in a relatively short period of time. In contrast to the persistent flapping analyzed in Section 4.2, these routing changes occur at a low enough rate to span multiple events. For example, a prefix may have a long-term instability due to flaky equipment that fails every few minutes, falling outside of our 70-second window for grouping BGP updates into events. Even if the equipment fails at a higher rate, the BGP updates may be suppressed periodically due to route-flap damping [5], leading to multiple events. Identifying these slowly *frequently flapping* prefixes is important for addressing long-term reachability problems and for reducing the number of BGP updates the routers need to handle.

To identify frequently flapping prefixes, we group events for the same destination prefix that occur *close together in time* (with an inter-arrival time less than $thresh_T$), and flag cases where the *number of events exceeds a predefined threshold* (max_count). We implement this heuristic by keeping track of each prefix that has had an event in the last $thresh_T$ seconds, along with the time of the last event and a count of the total number of events. Upon learning about a new event from RouteTracker, we check if the prefix has experienced an event in the last $thresh_T$ seconds and update the timestamp and counter values; once the counter exceeds max_count, we generate a report.

Since route changes can happen on virtually any timescale, the parameters $thresh_T$ and max_count

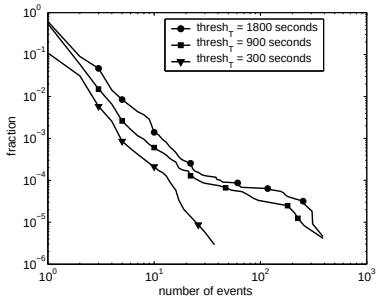

Figure 8: CCDF of the number of events per cluster for event correlation across time

should be set to highlight the most unstable prefixes without generating an excessive number of reports. Figure 8 shows the complementary cumulative distribution of the number of events per cluster over our eight-week measurement period. For all three values of $thresh_T$, more than 99% of the clusters have fewer than ten events; still, a small number of very large clusters exist. Having a very small $thresh_T$ might cause our system to overlook some unstable prefixes with a long cycle between routing changes. For example, a prefix that has a routing change every ten minutes would not be detected by a $thresh_T$ of 300 seconds. Based on the results in Figure 8, we assign $thresh_T$ to 900 seconds and max_count to 10 to draw attention to the small number of very unstable prefixes.

In our analysis, the percentage of events caused by frequently flapping prefixes varies from day to day from a low of 0.41% to a high of 32.78%, with an average of 3.38%. Most of these events are in category "loss/gain of reachability." We believe that frequent flapping tends to originate near the destination, making these instabilities visible to other ASes. To validate our inferences, we applied our heuristic for identifying frequently flapping prefixes to the BGP data from RouteViews [24]. For the week of September 26 to October 2, 2004, all 35 prefixes we identified were also flapping frequently in at least one other vantage point in the RouteViews data. Whether (and how) operators react to frequently flapping prefixes depends on the network responsible for the problem. If the frequent flapping comes from one of the ISP's own customers, the operators may be able to work with the customer to identify and fix the problem. If the flapping comes directly from a peer network (or one of the peer's customers), the operators may contact the peer to request that the peer address the problem.

6.2 Disruptions Affecting Multiple Prefixes

A single disruption (such as a link failure or a policy change) may affect multiple prefixes in a similar way, in a very short period of time. Grouping these prefixes together magnifies the visibility of the common effects and substantially reduces the number of reports for the operators. The five categories identified in Section 5.2 provide an effective way to identify prefixes affected in a "similar way." In addition, we also consider whether the border routers changed from a better route to a worse route, a worse route to a better route, or between two equally-good routes, in terms of the first six steps of the decision process in Table 1. This distinction gives us insight into whether the old route was withdrawn (or replaced by a less-attractive route), the new route recently appeared (or was replaced by a more-attractive route), or the router switched between two comparable routes (*e.g.*, because of a change in the IGP path costs).

In particular, we group events for different destination prefixes that (i) belong to the same category (using the taxonomy from Section 5.2), (ii) undergo the same kind of transition (from better to worse, or worse to better), and (iii) start no more than $thresh_P$ seconds after the first event. We consider the start time of the events because the first update is most likely to be directly triggered by the network event. We implement this heuristic by keeping track of the identifying information for each cluster (*i.e.*, the event category and the kind of transition) as well as the time of the first event and a count of the number of events. Upon generating a new event, we check if the event matches with the identifying information and arrives within $thresh_P$ seconds after the first event in the cluster. The correlation process adopts a clustering algorithm similar to those used in previous BGP root-cause analysis studies [6, 8, 13].

Setting $thresh_P$ too small runs the risk of splitting related events into two clusters. If a network disruption affects a large number of prefixes, the effects could easily spread over several tens of seconds. For example, a BGP session failure or hot-potato routing disruption that affects tens of thousands of prefixes requires the router to send numerous update messages, which could easily take up to a minute [28]. To account for these effects, we carefully select a value of 60 seconds for $thresh_P$ after a study of the duration traditional routing changes (*e.g.*, session resets) normally take to affect all of their related prefixes. Since $thresh_P$ is used to compare the start times of the two events, our heuristic cannot assume that a cluster is complete once the current time (the time of newly arrived BGP update in the system) is $thresh_P$ after the time of the first event in the cluster since an event may still be "in progress." Knowing that an event lasts at most the convergence timeout (from Section 4.2), in our heuristic, each cluster waits for a total of $thresh_P + convergence\_timeout$ to ensure that no ongoing, correlated events should be included in the cluster. In total, then, our heuristic waits for 660 seconds before declaring a cluster complete.[4]

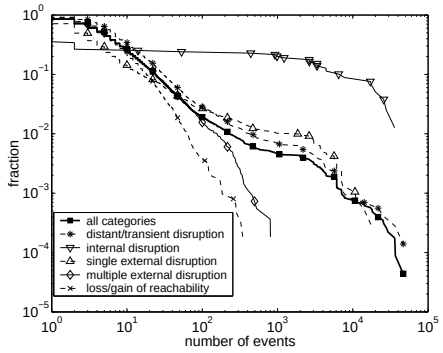

Figure 9: CCDF of the number of event per cluster for event correlation across prefixes

Figure 9 shows the effectiveness of clustering in combining related events. The graph plots the complementary cumulative distribution of the number of events per cluster over the eight-week period, on a log-log scale. Although 99% of the clusters have less than a hundred events (as shown in the "all categories" curve), a few clusters have a tremendous number of events. Meanwhile, the curves for different categories of events have distinctive characteristics. The categories "multiple external instability" and "loss/gain of reachability" have much smaller clusters, while the other three categories have some very large clusters with tens of thousands of affected prefixes. The categories "internal disruption" and "single external disruption" tend to have larger clusters than the other categories. Next, we show that these very large clusters stem from hot-potato routing changes and eBGP session resets, respectively.

6.2.1 Hot-Potato Changes

According to the BGP decision process in Table 1, a router selects among multiple equally good BGP routes (*i.e.,* routes that have the same local preference, AS path length, origin type, MED value, and eBGP vs. iBGP learned) the one with the smallest IGP cost. Such routing practice is called *hot-potato* routing [28]. An IGP topology change can trigger routers in a network to select a different *equally good* BGP route for the same prefix, and these changes may affect multiple prefixes. This section describes the routing disruptions caused by these hot-potato changes.

"Hot-potato" changes only affects the egress points each router selects for the prefixes. As the event classification in Section 5.2, it results in "internal disruptions" to the network. After the correlation process, the event cluster in category "internal disruption" magnifies the impact of the "hot-potato" changes. When these kinds of disruptions occur, the operators need to know which routers and prefixes are affected to gauge the significance of the event. Such information can be obtained by comparing the old and new r-vectors for all of the events in the cluster because each element in the r-vector carries the next-hop address for the corresponding router.

A previous study [28] proposed a heuristic for identifying hot-potato routing changes at a single router, based on a single stream of BGP updates from that router and data from an IGP topology monitor. Applying this technique to specific ingress routers allowed us to make direct comparisons between the two approaches. For the period from August 16 to September 30, 2004, over 95% of the large clusters (*i.e.,* clusters with more than 1000 events) of internal disruptions identified by our system are also identified using the technique in [28]. Inspecting the other 5% of cases in more detail, we discovered that these clusters corresponded to the restoration of a link in the network, where the failure had caused a previous hot-potato routing change that was detected using both techniques. As such, we believe that these disruptions are hot-potato routing changes that were not detected by the heuristic in [28].

6.2.2 eBGP Session Resets

The failure or recovery of an eBGP session can cause multiple events that affect the eBGP-learned routes from one neighbor at a single border router. Upon losing eBGP connectivity to a neighbor, a border router must stop using the routes previously learned from that neighbor and switch to less-attractive routes. The border router may switch to an eBGP-learned route from a different neighbor, if such a route exists; this would result in an "external path change" for the destination prefix. Alternatively, the router may have to switch to an iBGP-learned route from a different border router; this would result in a "loss of egress point" for the destination prefix. When the session recovers, the border router learns the BGP routes from the neighbor and switches back to the eBGP-learned routes advertised by this neighbor for one or more destination prefixes (causing either an "external path change" or a "gain of egress point").

To identify a session failure, we first group events that (i) belong to the category "single external disruption," (ii) have an *old* route with the same border router and neighbor (*i.e.,* the same $R_p^{j,old}$), (iii) have a routing change that goes from better to worse, and (iv) occur close together in time. However, this is not enough to ensure that the session failed, unless the router has stopped using most (if not all) of the routes previously learned from that neighbor. As such, we also check that the number of prefixes using the neighbor has decreased dramatically.[5] Similarly, to identify a session recovery, we first group events that (i) belong to the category "single external disruption," (ii) have a *new* route with the same border

router and neighbor (*i.e.*, the same $R_p^{j,old}$), (iii) have a routing change that goes from worse to better, and (iv) occur close together in time, and also involve a significant increase in the number of prefixes associated with that neighbor, back to the expected level.

Applying our heuristic to the "single external disruption" clusters that contain more than 1000 events, we found that 95.7% of these large clusters were linked to an eBGP session going up or down. To validate our inferences, we consulted the syslog data [16], which reports when the status of a BGP session changes. The syslog data confirmed more than 95% of our inferences. Our inferences not only captured all of the resets in syslog but identified a few disruptions that were not reported by syslog. Interestingly, we sometimes found that our analysis suggests that the session failure occurred up to ten seconds *before* the entry in the syslog data. After checking for possible timing discrepancies between the BGP and syslog data, we speculate that the remote AS is shutting down the BGP session in a graceful manner by first *withdrawing* all of the routes before actually disabling the session. This practice highlights the importance of using an algorithm such as ours even when syslog data are available.[6] A complete loss of the routes from a neighbor does *not* necessarily arise only from a session failure. Instead, the neighbor's router may be reconfigured with a new policy (*e.g.*, that withdraws the previous routes) or lose connectivity to other routers in its own network. These kinds of disruptions could have a significant impact on traffic inside an AS, and would not generate a syslog report. The influence of large disruptions on the traffic is explored in more detail in the next section.

7 Estimating Traffic Impact

We now describe the final component of the system—TrafficMeter which allows us to estimate the traffic impact of the routing disruptions produced by the Event-Correlator. Although the traffic volume on a link typically varies gradually across days and weeks, sudden changes in traffic can lead to congestion in some parts of the network. A recent study [26] shows BGP routing disruptions are responsible for many of the largest traffic shifts in backbone networks. Below we first discuss how we compute traffic weights to estimate the impact on traffic and then focus on two types of routing disruptions with the most impact.

7.1 Computing Traffic Weights

TrafficMeter aggregates the Netflow data [1] collected on the outgoing links to compute prefix-level traffic statistics. For each destination prefix, we define a *traffic weight* that corresponds to the percentage of traffic destined to that prefix across the overall traffic volume in the network. In essence, the weight corresponds to the relative popularity of the prefix. Since the proportion of traffic destined to each prefix changes over time, we compute the weights over a sliding time window (*e.g.*, the last month). The weights allow us to estimate the potential impact of a cluster of routing events by considering the sum of the weights for all prefixes in the cluster. Although the weights do not capture the variations in traffic per prefix across time and location, they do provide a simple way to flag routing disruptions that affect clusters of prefixes that attract a high volume of traffic.

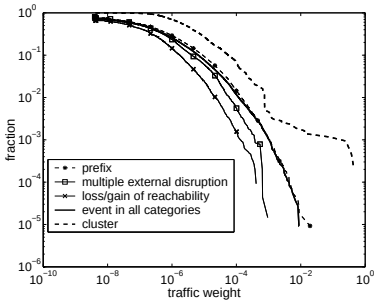

Figure 10: CCDF of traffic weight

In Figure 10, we plot the complementary cumulative distribution of traffic weight of a prefix, an event, and an event cluster over the eight-week period of our study. The "prefix" curve shows the significant differences in popularity of the prefixes, consistent with previous studies [11, 22]. Interestingly, the "event in all categories" curve looks largely the same, suggesting that routing events affect prefixes across the entire range of popularities. This occurs because the many events in categories "distant/transient disruption," "single external disruption," and "internal disruption" tend to affect a wide range of destination prefixes, largely independent of their popularity; the curves for these three categories of events are not shown, as they look almost identical to the "prefix" and "event in all categories" curves. In contrast, the curves for events in categories "multiple external disruption" and "loss/gain of reachability" suggest that these events tend to involve prefixes that receive less traffic.

The "cluster" curve plots the distribution of traffic weight across the event clusters. As expected, a cluster tends to have a large traffic weight since it combines one or more related events. The tail of the curve suggests that a small number of clusters are responsible for a significant portion of the large traffic shifts. Meanwhile, our results reveal that these "significant" clusters have a large number of events, implying the routing change affects many prefixes. Our system observes a few dozen such large clusters each day and highlights them for the

network operators for their attention. We use the threshold of 1% for traffic weight to signal significant routing disruptions, since the vast majority of clusters fall below that threshold. This avoids operators focusing their attention on the many BGP disruptions that affect a very small fraction of the traffic.

7.2 Disruptions With Large Weights

We now discuss our empirical findings using TrafficMeter based on our eight weeks of measurement data. Interestingly, most big events in terms of the amount of traffic weight are single external disruptions and internal disruptions. Thus, we focus on those in Figure 11 showing the duration of a routing disruption relative to the corresponding traffic weight of the affected prefixes for clusters with traffic weight larger than 1%. On average, internal disruptions (*e.g.*, hot potato changes) result in larger traffic weights than single external disruption (*e.g.*, session resets), because internal routing disruptions usually affect multiple locations. They also appear to have longer durations than single external disruptions. Long-lived events allow operators to adapt routing configurations as needed to alleviate possible network congestion. Our tool highlights only a few critical events which are both long-lived and expected to affect a large amount of traffic. This helps focus operators' attention on routing disruptions where mitigation actions, such as tuning the routing protocol configuration, might be necessary.

Figure 11 also shows that our tool captures some large disruptions that are short-lived, lasting 30 seconds to a few minutes. In addition to most of the "single external disruption" points in the graph, these short-lived disruptions include many large clusters in the "distant/transient disruption"; this category accounts for 78.8% of all event clusters with traffic weight higher than 1%. These clusters involve events that start and end with the same route vectors, with some sort of transient disruption in between. Although short-lived traffic shifts do not have a sustained impact on network load, users may encounter brief periods of degraded performance that could be traced to these disruptions. Interestingly, these short-lived traffic shifts are extremely difficult to detect using conventional measurement techniques, such as SNMP and Netflow, that aggregate traffic statistics on the timescale of minutes. In contrast, our troubleshooting system can identify short-lived routing disruptions that may have large effects on user performance.

8 System Evaluation

In this section, we demonstrate that our system imposes a small amount of memory and CPU processing overhead to run in real time on a commodity computing platform.

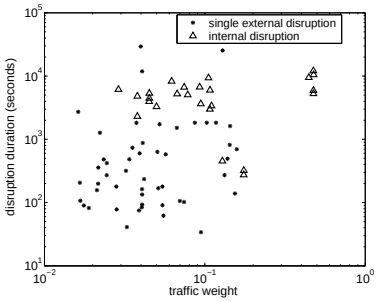

Figure 11: Routing disruption durations vs. traffic weights

Throughout the evaluation of our system on eight weeks of data, the system memory footprint never exceeded 900 Megabytes and every interval of 70 seconds of BGP updates was processed in less than 70 seconds.

We characterize the system performance through an off-line emulation over the past measurement data. Due to operational concerns, our system could not access the collected data in real-time. Instead, we stored the measurements locally and replayed the data in our tool. We ran our tool on a Sun Fire 15000 equipped with several 900 MHz Ultrasparc-II processors. Only one processor was used during the experiments. We evaluate the system using two metrics: *memory usage* and *execution speed*.

8.1 Memory Usage

The memory usage in our troubleshooting system consists of two parts: *static* usage and *dynamic* usage. The static memory is allocated to store the best route for each border router and destination prefix. In the core of today's Internet, each router learns reachability information for about 160,000 prefixes (also confirmed by RouteViews [24]). The total static memory usage in our system is about 600 Megabytes.

Dynamic memory, on the other hand, is allocated to maintain the data structures continuously created in response to the arrival of BGP updates. The essential data objects kept in the system are clusters, whose memory are dynamically allocated and reclaimed during the process as discussed in Section 6. In processing the eight weeks of measurement data, the dynamic memory footprint of the system never exceeded 300 Megabytes.

8.2 Execution Speed

We measure how quickly the system processes the BGP updates. Because the progression of each BGP update in the system varies depending on the expiration condition of several timers, we have conducted the experiment for each BGP update sequence within a fixed time interval called *epoch*, rather than characterizing the execution la-

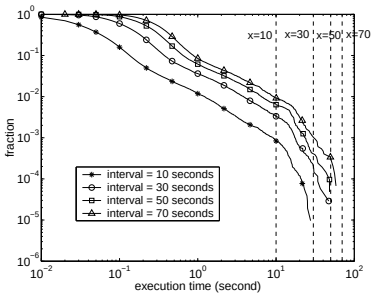

Figure 12: System execution speed

tency of each individual BGP update. During each test, we randomly selected a starting point in the eight-week BGP update sequence and then divided the subsequent BGP update stream into non-overlapping epochs. Then we measured the execution time for each epoch of a fixed epoch interval. We varied the epoch interval among the values of 10, 30, 50, 70 seconds. Because the machine is a time-sharing system, we ran each experiment three times to ensure the accuracy of the measurement results; we saw virtually no variation in the results across the three experiments.

Figure 12 shows the complementary cumulative distribution of the execution time for each of the four epoch intervals. As shown in the graph, the execution of nearly every epoch was completed within the epoch interval. For example, the curve for a ten-second epoch interval shows that more than 99% of epochs could be processed within one second; however 0.1% of the epochs required more than ten seconds to complete. Our system occasionally lags behind the arrival of BGP updates, due to the bursty arrival pattern of BGP updates. Our data show that, while the average number of BGP updates per second is well below 100 (which corresponds to about 30 Kbps data rate), the maximum number of BGP updates received in our system in one second could well exceed 10,000 (which corresponds to 3 Mbps data rate).

Despite the existence of execution lags, for an epoch interval of 30 seconds, its percentage becomes much smaller (0.01%) by smoothing the BGP update bursts with a longer interval. The execution lag is completely eliminated when we set the epoch interval to 70 seconds; that is, every interval of 70 seconds worth of BGP updates was completely processed in less than 70 seconds. We believe the occasional execution lag is acceptable. Recall that each event is identified only if at least a period of event timeout elapses after the arrival of the last BGP update in the event. Typically the timeout value is a few tens of seconds (70 seconds, in our experiments). That is, even with instantaneous processing, each BGP update would have to wait for at least 70 seconds before a report is generated for the network operators. As such, smoothing the processing of BGP updates over a few tens of seconds does not introduce a problem.

9 Related Work

There is a large body of literature on characterizing BGP data using passive monitoring [3, 4, 9, 22, 29] as well as active route injection [17]. Our study is also preceded by several recent efforts [6, 8, 13, 15, 30] to identify the location and cause of routing changes by analyzing BGP update messages along three dimensions: time, views, and prefixes. Our work is similar in that we analyze BGP data along the same dimensions to group related routing changes. However, we focus on organizing large volumes of BGP updates seen in a single AS in real time into a small number of reports belonging to categories directly useful to operators to help mitigate the problems.

In analyzing BGP data collected from multiple vantage points within a single AS, our work is similar to the BorderGuard [12] study that identifies inconsistent routing advertisements from peers. In contrast, we classify *all* routing changes seen by the border routers into useful categories. The work in [27] presents a strawman proposal where each AS collects BGP data from its border routers as part of an end-to-end service for identifying the location and cause of routing changes. Each AS uses the data to detect and explain its own *internal* routing changes, rather than trying to detect and diagnose inter-domain routing events. Recent work [23] has considered how to detect network anomalies through a joint analysis of traffic and routing data. This work looks for significant changes in both the volume of traffic and the number of update messages, without delving in to the details about the specific destination prefixes and event types involved.

10 Conclusion and Future Work

We have presented the design and evaluation of an online system for identifying important BGP routing changes in an IP network. Using the concise r-vector data structure to capture BGP routing changes, we identified five categories of BGP routing disruptions that vary in the severity of the impact on the traffic. Applying the tool to eight weeks of routing and traffic data from a tier-1 ISP network, we identified several ways for operators to improve the routing stability of the network. Despite having route-flap damping features enabled on all of the routers, our tool surprisingly discovered a large number of updates from persistently flapping prefixes and identified three causes. Meanwhile, we found that hot-potato routing changes and eBGP session resets were responsible for many of the large routing disruptions.

In our ongoing work, we are extending the system to

use fine-grained traffic data collected at the ingress points for more precise estimates of the traffic impact. We also plan to explore routing architectures, operational practices, and protocol enhancements that reduce the likelihood and impact of the routing disruptions associated with hot-potato changes and eBGP session resets. Finally, we plan to explore automated techniques for responding to disruptions by reconfiguring the routing protocols to improve network performance.

Acknowledgments

Thanks to Alex Gerber, Joel Gottlieb, Renata Teixeira, and Jen Yates for their help with the measurement data. Thanks also to Mike Bailey, Jay Borkenhagen, Jaideep Chandrashekar, Evan Cooke, Nick Feamster, Ningning Hu, Ramana Kompella, Dan Pei, Aman Shaikh, Kang Shin, and the anonymous reviewers for their detailed comments on an earlier version of the paper. We would also like to thank our shepherd, Ramesh Govindan.

References

[1] Netflow. `http://www.cisco.com/warp/public/732/Tech/nmp/netflow/index.shtml`.

[2] RIPE Routing-WG recommendations for coordinated route-flap damping parameters, Oct. 2001. Document ID: ripe-229, `http://www.ripe.net/ripe/docs/routeflap-damping.html`.

[3] C. LABOVITZ, R. MALAN, AND F. JAHANIAN. Internet routing instability. In *Proc. ACM SIGCOMM* (1997).

[4] C. LABOVITZ, R. MALAN, AND F. JAHANIAN. Origins of Internet routing instability. In *Proc. IEEE INFOCOM* (1999).

[5] C. VILLAMIZAR, R. CHANDRA, AND R. GOVINDAN. BGP route flap damping. *RFC 2439* (1998).

[6] CAESAR, M., SUBRAMANIAN, L., AND KATZ, R. H. Towards localizing root causes of BGP dynamics. Tech. Rep. CSD-03-1292, UC Berkeley, November 2003.

[7] CHANG, D., GOVINDAN, R., AND HEIDEMANN, J. An empirical study of router response to large BGP routing table load. In *Proc. Internet Measurement Workshop* (November 2002).

[8] CHANG, D.-F., GOVINDAN, R., AND HEIDEMANN, J. The temporal and topological characteristics of BGP path changes. In *Proc. IEEE ICNP* (November 2003).

[9] D. WETHERALL, R. MAHAJAN, AND T. ANDERSON. Understanding BGP misconfigurations. In *Proc. ACM SIGCOMM* (2002).

[10] E. CHEN AND T. BATES. An application of the BGP community attribute in multi-home routing. *RFC 1998* (1996).

[11] FANG, W., AND PETERSON, L. Inter-AS traffic patterns and their implications. In *Proc. IEEE Global Internet* (December 1999).

[12] FEAMSTER, N., MAO, Z. M., AND REXFORD, J. BorderGuard: Detecting cold potatoes from peers. In *Proc. Internet Measurement Conference* (October 2004).

[13] FELDMANN, A., MAENNEL, O., MAO, Z. M., BERGER, A., AND MAGGS, B. Locating Internet routing instabilities. In *Proc. ACM SIGCOMM* (August 2004).

[14] LABOVITZ, C., AHUJA, A., BOSE, A., AND JAHANIAN, F. Delayed Internet routing convergence. *IEEE/ACM Trans. on Networking 9*, 3 (June 2001), 293–306.

[15] LAD, M., NANAVATI, A., MASSEY, D., AND ZHANG, L. An algorithmic approach to identifying link failures. In *Proc. Pacific Rim Dependable Computing* (2004).

[16] LONVICK, C. The BSD syslog protocol. RFC 3164, August 2001.

[17] MAO, Z. M., BUSH, R., GRIFFIN, T. G., AND ROUGHAN, M. BGP beacons. In *Proc. Internet Measurement Conference* (2003).

[18] MCPHERSON, D., GILL, V., WALTON, D., AND RETANA, A. Border gateway protocol (BGP) persistent route oscillation condition. RFC 3345, August 2002.

[19] N. DUFFIELD, C. LUND, AND M. THORUP. Estimating flow distributions from sampled flow statistics. In *Proc. ACM SIGCOMM* (2003).

[20] O. MAENNEL AND A. FELDMANN. Realistic BGP traffic for test labs. In *Proc. ACM SIGCOMM* (2002).

[21] REKHTER, Y., LI, T., AND HARES, S. A Border Gateway Protocol 4 (BGP-4). Internet Draft draft-ietf-idr-bgp4-26.txt, work in progress, Oct. 2004.

[22] REXFORD, J., WANG, J., XIAO, Z., AND ZHANG, Y. BGP routing stability of popular destinations. In *Proc. Internet Measurement Workshop* (2002).

[23] ROUGHAN, M., GRIFFIN, T., MAO, Z. M., GREENBERG, A., AND FREEMAN, B. Combining routing and traffic data for detection of IP forwarding anomalies. In *Proc. SIGCOMM Network Troubleshooting Workshop* (2004).

[24] ROUTEVIEWS. http://www.routeviews.org/.

[25] T. GRIFFIN AND G. WILFONG. Analysis of the MED Oscillation problem in BGP. In *Proc. IEEE ICNP* (November 2002).

[26] TEIXEIRA, R., DUFFIELD, N., REXFORD, J., AND ROUGHAN, M. Traffic matrix reloaded: Impact of routing changes. In *Proc. Passive and Active Measurement Workshop* (March/April 2005).

[27] TEIXEIRA, R., AND REXFORD, J. A measurement framework for pin-pointing routing changes. In *Proc. SIGCOMM Network Troubleshooting Workshop* (September 2004).

[28] TEIXEIRA, R., SHAIKH, A., GRIFFIN, T., AND REXFORD, J. Dynamics of hot-potato routing in IP networks. In *Proc. ACM SIGMETRICS* (2004).

[29] WANG, L., ZHAO, X., PEI, D., BUSH, R., MASSEY, D., MANKIN, A., WU, S. F., AND ZHANG, L. Observation and analysis of BGP behavior under stress. In *Proc. Internet Measurement Workshop* (2002).

[30] WONG, T., JACOBSON, V., AND ALAETTINOGLU, C. Making sense of BGP, Feb. 2004. NANOG presentation.

Notes

[1] Since the routes from $c1$ and $c3$ have the same IGP path cost, the router performs an arbitrary tiebreak in the last step in Table 1.

[2] This would require either (i) extending today's commercial routers to provide a feed of updates for each eBGP session or (ii) deploying packet monitors on all peering links to capture BGP update messages.

[3] The requirement of having at least one eBGP-learned route for the prefix is necessary to distinguish the "multiple external disruption" category from the "loss/gain of reachability" category.

[4] Our system can generate reports about large clusters (*i.e.*, when the count of events exceeds a threshold) before the 660-second timer expires, to allow operators to react more quickly to significant events.

[5] In theory, we could check that the number drops to zero. However, maintaining the prefix count for each session in real time (*e.g.*, updating whenever a prefix is withdrawn or advertised) introduces substantial overhead. Instead, noticing that the prefix count stays stable, we sample the count every two hours and maintain a moving average. We flag cases where the number of events in the cluster is more than 90% of the average number of prefixes associated with that session.

[6] Furthermore, syslog uses UDP as its transport layer and therefore its data can be lost during delivery.