Effective Diagnosis of Routing Disruptions from End Systems

Ying Zhang
University of Michigan

Z. Morley Mao
University of Michigan

Ming Zhang
Microsoft Research

Abstract

Internet routing events are known to introduce severe disruption to applications. So far effective diagnosis of routing events has relied on proprietary ISP data feeds, resulting in limited ISP-centric views not easily accessible by customers or other ISPs. In this work, we propose a novel approach to diagnosing significant routing events associated with any large networks from the perspective of end systems. Our approach is based on scalable, collaborative probing launched from end systems and does not require proprietary data from ISPs. Using a greedy scheme for event correlation and cause inference, we can diagnose both interdomain and intradomain routing events. Unlike existing methods based on passive route monitoring, our approach can also measure the impact of routing events on end-to-end network performance. We demonstrate the effectiveness of our approach by studying five large ISPs over four months. We validate its accuracy by comparing with the existing ISP-centric method and also with events reported on NANOG mailing lists. Our work is the first to scalably and accurately diagnose routing events associated with large networks entirely from end systems.

1 Introduction

The end-to-end performance of distributed applications and network services is known to be susceptible to routing disruptions in ISP networks. Recent work showed routing disruptions often lead to periods of significant packet drops, high latencies, and even temporary reachability loss [1, 2, 3, 4]. The ability to pinpoint the network responsible for observed routing disruptions is critical for network operators to quickly identify the problem cause and mitigate potential impact on customers. In response, operators may tune their network configurations or notify other ISPs based on the inferred origin location of the routing disruption: internal networks, border routers, or remote networks. They may also find alternate routes or inform impacted customers about destinations expected to experience degraded performance.

From the perspective of end users, the ability to diagnose routing disruptions also provides insight into the reliability of ISP networks and ways to improve the network infrastructure as a whole. Knowing which ISPs should be held accountable for which routing disruptions helps customers assess the compliance of their service-level agreements (SLAs) and moreover provides strong incentives for ISPs to enhance their service quality.

Past work on diagnosing routing events has relied on routing feeds from each ISP. These techniques have proven to be effective in pinpointing routing events across multiple ISPs [5] or specific to a particular ISP [6]. However, given that most ISPs are reluctant about revealing details of their networks, they normally keep their routing feeds publicly inaccessible. Today, the largest public routing data repositories, RouteViews and RIPE, receive data from only around 154 ISPs [7, 8], in most cases with at most one feed from each AS. These feeds have been shown to be insufficient to localize routing events to a particular ISP [9]. As a result, customers are in the dark about whether their service providers meet their service agreements. Similarly, ISPs have limited ways to find out whether the problems experienced by their customers are caused by their neighbors or some remote networks. They usually have to rely on phone calls or emails [10] to perform troubleshooting.

Motivated by the above observations, we aim to develop new techniques for diagnosing routing events from end systems residing at the edge of the Internet. Our approach differs markedly from existing work on pinpointing routing events by relying only on probes launched from end-hosts and not requiring any ISP proprietary information. Using active probing on the data plane, our system can in fact more accurately measure the performance of actual forwarding paths rather than merely knowing the expected routes used based on routing advertisements. Furthermore, our techniques can be easily applied to many different ISPs without being restricted to any particular one. This is especially useful for diagnosing inter-domain routing events which often require cooperation among multiple ISPs. Our inference results can be made easily accessible to both customers and ISPs who need better visibility into other networks. This is also helpful for independent SLA monitoring and management of routing disruptions. In addition, end system probing can be used for both diagnosing and measuring the performance impact of routing events. It offers us a unique perspective to understand the impact of routing events on end-to-end network performance.

In this paper, we consider the problem of diagnosing routing events for any given ISP based on end system probing. Realizing that identifying the root cause of

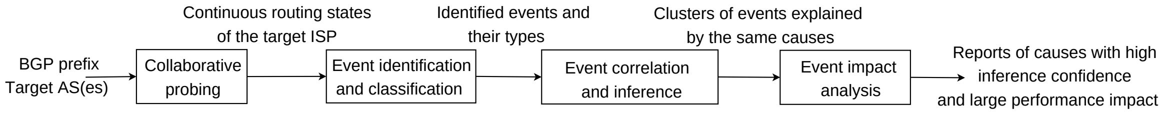


Figure 1: System Architecture

routing events is intrinsically difficult as illustrated by Teixeira and Rexford [9], we focus on explaining routing events that the ISP should be held accountable for and can directly address, *e.g.,* internal routing changes and peering session failures. In essence, we try to tackle the similar problem specified by Wu *et al.* [6] without using ISP's proprietary routing feeds. Given that end systems do not have any direct visibility into the routing state of an ISP, our system overcomes two key challenges: i) discovery of routing events that affect an ISP from end systems; and ii) inference of the cause of routing events based on observations from end systems. We present the details of our approach and its limitations in terms of coverage, probing granularity, and missed routing attributes in §3.

We have designed and implemented a system that diagnoses routing events based on end system probing. Our system relies on collaborative probing from end systems to identify and classify routing events that affect an ISP. It models the routing event correlation problem as a bipartite graph and searches for plausible explanation of these events using a greedy algorithm. Our algorithm is based on the intuition that routing events occurring close together are likely explained by only a few causes, which do not create many inconsistencies. We also use probing results to study the impact of routing events on end-to-end path latency.

We instantiate our system on PlanetLab and use it to diagnose routing events for five big ISPs over a period of four months. Although each end-host has only limited visibility into the routing state of these ISPs, our system can discover many significant routing events, *e.g.,* hot-potato changes and peering session resets. Compared to existing ISP-centric method, our approach can distinguish internal and external events with up to 92.7% accuracy. Our system can also identify the causes for four out of the six disruptions reported from NANOG mailing lists [10] during that period.

We summarize our main contributions. Our work is the first to enable end systems to scalably and accurately diagnose causes for routing events associated with large ISPs without requiring access to any proprietary data such as real-time routing feeds from many routers inside an ISP. Unlike existing techniques for diagnosing routing events, our approach of using end system based probing creates a more accurate view of the performance experienced by the data-plane forwarding path. Our work is an important first step to enable diagnosis of routing disruptions on the global Internet accounting for end-to-end performance degradations.

2 System Architecture

We present an overview of our system in this section. To diagnose routing events for any given ISP (which we call **a target ISP**), our system must learn the continuous routing state of the ISP. Based on the change in routing state, it identifies and classifies individual routing events. Because a single routing disruption often introduces many routing events, our system applies an inference algorithm to find explanations for a cluster of events occurring closely in time. It then uses the latency measurements in the probes to quantify the impact of these routing events. As shown in Figure 1, our system is composed of four components:

Collaborative probing: This component learns the routing state of a given ISP via continuous probing from multiple end systems. Given the large number of destinations on the Internet, the key challenge is to select an appropriate subset to ensure coverage and scalability.

Event identification and classification: This component identifies routing events from a large number of end-system probes. These events are then classified into several types based on the set of possible causes, *e.g.,* internal changes, peering failures, or external changes.

Event correlation and inference: This component searches for plausible explanation for routing events. Although each routing event may be triggered by many possible causes, we seek to identify a small set of causes that can explain all the events occurring close in time. We model the inference problem as a bipartite graph and solve it with a greedy algorithm.

Event impact analysis: This component extracts latency information from end-system probes. It enables us to study the impact of routing events on path latency according to the cause of events and the impacted ISPs. Note that this information is not readily available in routing feeds used in previous work on routing diagnosis.

3 Collaborative Probing

For a target ISP, we need to know its routing state to identify and diagnose its routing events. Unlike previous work that uses many routing feeds from a single ISP [6],

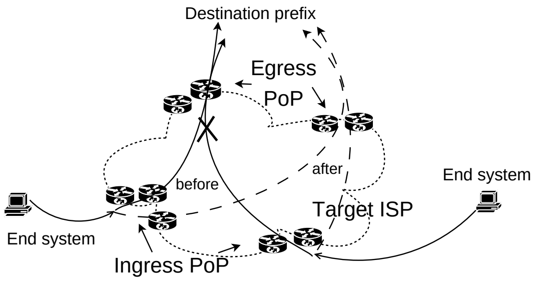


Figure 2: Collaborative probing to discover routing events.

our system relies on end systems that do not have any direct visibility into ISP's routing state. Note that it is important to obtain a comprehensive view of the routing state across major Points of Presence (PoPs) of the target ISP in order to diagnose routing events associated with the ISP. Utilizing public routing repositories is insufficient due to only one or at most two feeds from each ISP, in addition to issue of a lack of real-time data feeds. The key question in our design is how to learn the routing state of an ISP from end-system probing alone.

3.1 Learning routing state via probing

A router's routing table contains the traffic forwarding information, *e.g.,* the next hop, based on the destination prefix. Although an end system may not have direct access to the routing tables, it could learn this next hop information using *traceroute* if the forward path from the host to the destination happens to traverse the router. As illustrated in Figure 2, traceroute probing from two end systems to one particular destination experiences egress PoP shifts due to the target ISP's internal disruption. Ideally, we can learn the next hop from any router to any destination by probing from an appropriate source. This is not always be possible because we may not have access to such a source or the router may not respond to our probes.

We focus on diagnosing inter-domain routing events that affect a target ISP. We aim to find explanations for events that the ISP should be held accountable for and can directly address, *e.g.,* internal routing changes and peering session failures. For internal or intra-domain routing events it is obvious which ISP should take responsibility for them. Therefore, we do not focus on constructing detailed intra-domain routing tables. Instead, we keep track of the inter-domain routing tables (BGP tables) of each major PoP within the ISP.

There are three challenges associated with constructing BGP tables. First, given a limited set of end systems, the system attempts to obtain as many routes between PoP-prefix pairs (PoP to destination prefix) as possible. Second, end systems have limited resources (CPU and network), and our system must have low probing overhead. Third, probing needs to be launched frequently to accurately track the dynamic routing state.

To address the first two challenges, we devise a scheme to select an appropriate set of destinations for each end system to probe. We start with a set of prefixes extracted from BGP tables. Each end system acquires its own routing view by conducting traceroute to one IP in each of these prefixes. Using the existing method developed in Rocketfuel [11], we can infer whether each traceroute probe goes through the target ISP and the PoPs traversed. Combining the routing views from all the end systems, we obtain a complete set of PoP-prefix pairs visible to our system. We then try to select a minimum set of traceroute probes that can cover all the visible PoP-prefix pairs with a greedy algorithm. At each step, we select a traceroute probe that traverses the maximum number of uncovered PoP-prefix pairs and remove these newly-covered pairs from the set of uncovered pairs. This process continues until there is no uncovered PoP-prefix pair left. The selection process has been shown to be effective in balancing between coverage and overhead [12]. Note that because ISP network topology and routing evolve over time, each end system periodically refreshes its routing view. Currently, this is done once a day to achieve a balance between limiting probing overhead and capturing long-term changes.

To address the third challenge, we developed a customized version of traceroute which enhances the probing rate by measuring multiple destinations and multiple hops in parallel up to a pre-configured maximum rate. To prevent our measurement results from being affected by load-balancing routers, all probe packets have the same port numbers and type of service value. With our improvement, all the end systems can finish probing their assigned set of destinations in roughly *twenty minutes*. This also means that our system can obtain a new routing state of the target ISP every twenty minutes, the details of which are shown in §6.

3.2 Discussion

Although learning an ISP's routing state via collaborative probing does not require any ISP proprietary information, it has three major limitations compared with direct access to BGP routing feeds: (i) given a limited number of end systems, we cannot learn the route for every PoP-prefix pair; (ii) given limited CPU and network resources at end systems, we cannot probe every PoP-prefix pair as frequently as desired. This implies we may miss some routing events that occur between two consecutive probes; and iii) we can only observe forwarding path changes but not other BGP attribute changes.

The first problem of coverage is a common hurdle for systems finding root causes of routing changes as described by Teixeira and Rexford [9]. They presented an idealized architecture for cooperative diagnosis which requires coverage in every AS. Similar to the work by Wu *et al.*, our work addresses a simpler problem of diagnosing routing changes associated with a large ISP but purely from end system's perspectives. Our ability to address this relies on the coverage obtained.

A straightforward solution to improving coverage is to use more end systems. In this paper, we use all the available PlanetLab sites (roughly 200) to probe five target ISPs. We will explain the detailed coverage results in §6. Note that a single major routing disruption near the target ISP, *e.g.*, a hot-potato change or a peering session failure, often introduces a large number of routing events and affects many different PoPs and prefixes. In §7, we will show that our system is able to correctly identify many such major disruptions despite covering only a subset of the affected PoP-prefix pairs. As future work, we plan to study how better coverage will improve our inference accuracy. Besides the coverage limitation, topology discovery could be affected by ISPs' ICMP filtering policy. Fortunately, we find this is performed mostly by ISPs on their edge routers connecting to customers, which has little impact on our inference.

We consider the second problem of limited probing frequency to be less critical. Our system focuses on diagnosing routing changes that are long-lived enough to warrant ISP's corrective action rather than transient ones that may repair by themselves quickly. Reporting every transient event may actually overwhelm ISP operators.

The third problem is more fundamental to systems that rely on end-system probing, given that BGP data can be inherently proprietary. This implies we might identify or locate a routing change but might not know *why* it occurs. We give an example of this in §5 where we cannot distinguish a route change triggered by different attribute changes. The focus of our work is on determining whether an ISP should be held accountable for a routing problem and providing useful hints for the ISP to diagnose it. We believe the responsible ISP can subsequently use its own data to perform root cause analysis.

4 Event Identification and Classification

In this section, we first describe how we identify individual routing events from the time sequence of routing state captured for a target ISP. We then present our event classification method based on likely causes.

4.1 Data processing

As explained in the previous section, we focus on the inter-domain routing state of the target ISP. Given a PoP-prefix pair, we identify the next hop and the AS path from the PoP to the destination prefix. The next hop can be either a PoP in the target ISP or another ISP. This implies that we need to extract the ISP and PoP information from end systems' traceroute probes.

A traceroute probe only contains the router's interface address along the forwarding path from the source to the destination. We map an IP address to a PoP in the target ISP using the existing tool based on DNS names (*undns*) [13]. For instance, 12.122.12.109 reverse-resolves to *tbr2-p012601.phlpa.ip.att.net*, indicating it is in the AT&T network, located in Philadelphia (phlpa). *undns* contains encoded rules about ISPs' naming conventions. For IP addresses not in the target ISP, we map them to ASes based on their origin ASes in the BGP tables [14]. One IP address may map to multiple origin ASes (MOAS) and we keep a set of origin ASes for such IP addresses. After performing IP-to-PoP and IP-to-AS mappings for each traceroute probe, we know the traversed PoPs in the target ISP and the AS path to the destination prefix. Given that errors in IP-to-AS and IP-to-PoP mappings are sometimes inevitable, we present a greedy algorithm that lowers the number of incorrect mappings by reducing total conflicts in event correlation and inference (§5).

Note that not all traceroute probes are used for routing event identification and classification. They may be discarded for several reasons:

Not traversing the target AS: Traceroute probes may not traverse the target ISP when the source hosts do not have up-to-date routing views or the probes are conducted during temporary routing changes. Such probes are discarded because they do not contribute any routing information about the target ISP.

Contiguous "*" hops: Traceroute paths may contain "*" hops when routers do not respond to probes due to ICMP filtering or rate-limiting. A "*" hop is treated as a wildcard and can map to any ISP or PoP. To simplify path matching for event identification, we discard traceroute containing two or more consecutive "*" hops.

Loops: Traceroute paths may contain transient loops that likely capture routing convergence. Such traceroute paths are not stable and somewhat arbitrary because they depend on the subtle timing when routers explore alternate paths. Since our goal is to infer the likely causes of routing events, we are interested in the stable paths before and after a routing event rather than the details of the transition. We discard traceroute paths that contain IP-level, PoP-level, or AS-level transient loops.

Some traceroute paths may contain loops that persist for more than 20 minutes. Since most routing convergence events last for several minutes [15], these loops are likely caused by routing misconfigurations [16] rather than unstable router state during convergence. We still make use of such traceroute paths after truncating their loops, since the partial paths represent stable paths.

4.2 Event identification and classification

We now describe how we identify inter-domain routing events that affect the target ISP from the continuous snapshots of routing state obtained from traceroute probes. An *inter-domain routing event* is defined as a path change from a PoP to a destination prefix, in which either the next hop or the AS path has changed. Since our system acquires a new routing state of the target ISP periodically, we can identify an event by observing a path change between the same source and destination in two consecutive measurements.

Given that there could be "*" hops and multiple-origin-ASes (MOAS) hops, we choose to be conservative in comparing two paths by trying to search for their best possible match. For instance, $path(A,\ *,\ C)$ is considered to match $path(A,\ B,\ C)$ because "*" can match any ISP or PoP. Similarly, a MOAS hop can match any AS in its origin AS set.

When observing path changes between two consecutive measurements, we classify them into three types according to their likely causes. The classification is motivated by our goal of inferring the causes of the changes relative to the target ISP.

Type 1: Different ingress PoP changes can be caused by routing events in the upstream ISPs, the target ISP, or downstream ISPs. Realizing it is difficult to enumerate all possible causes, we do not currently use them for event correlation and inference.

Type 2: Same ingress PoP but different egress PoP changes can be caused by internal disruptions in the target ISP (*e.g.*, hot-potato changes), failures on its border (*e.g.*, peering session reset), or external changes propagated to the target ISP (*e.g.*, prefix withdrawals).

Type 3: Same ingress PoP and same egress PoP changes are easier to deal with compared to the previous two types. They may involve internal PoP path changes, external AS path changes, or both. We will explain how to use such information for event correlation and inference in the next section.

5 Event Correlation and Inference

It is well known that a single major routing disruption often leads to a burst of routing events and affects many PoPs and prefixes simultaneously. Our goal is to diagnose which inter-domain routing events are triggered by those major disruptions that the target ISP should be held accountable for and can take action on.

In many cases, it is extremely difficult to infer the cause of an individual routing event because an event may be explained by many different causes. An obvious solution is to improve inference accuracy by correlating multiple "relevant" events together. However, the key question is how we can discover and make use of the relevancy among events.

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 the same AS
6. eBGP learned route over iBGP learned route
7. Lowest IGP cost (hot-potato)
8. Lowest router ID

Table 1: BGP decision process

5.1 Inference model

Before describing our inference model used for event correlation, we make an assumption that each routing event can be explained by only one cause. This is a standard assumption made in many existing work on root cause analysis [5, 9] and fault diagnosis [17]. Note that this assumption does not prevent us from inferring multiple simultaneous causes as long as the events triggered by different causes are independent.

We start by defining some terminology to facilitate our discussion. Since each event is identified by observing the change between two consecutive probes, we call the earlier path probe an **old path** and the later one a **new path**. We call the egress PoP on the old/new path the old/new egress respectively. In the previous section, we classify individual routing events into three types. Currently, we do not use the events of the first type for correlation because it is infeasible to enumerate all the possible causes for them. We identify all the possible causes for the latter two types of events based on how BGP selects a single best route for each prefix. When multiple routes are available, BGP follows the decision process in Table 1 to select the best one.

Same ingress PoP but different egress PoP changes can be triggered by a prefix withdrawal, a prefix announcement, or a change in any of the eight steps in Table 1. We ignore $Step_8$ since router ID rarely changes. $Step_6$ is irrelevant because both the old and the new egress use external paths. The following causes comprehensively cover all the remaining possibilities:

- A change in $Step_1$ is explained by either an *Old-Peering-Down* or a *New-Peering-Up*. The former implies the peering between the old egress and its neighbor AS is down. The latter means the peering between the new egress and its neighbor is up.
- A change in $Step_2$ can be explained by either an *Old-Lpref-Decrease* or a *New-Lpref-Increase*. The former implies the local preference ($Lpref$) at the old egress decreases. The latter implies the $Lpref$ at the new egress increases.

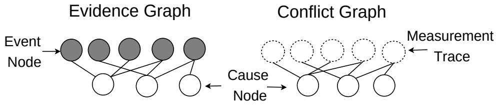


Figure 3: The bipartite graphs for cause inference

- A prefix withdrawal, an announcement, or a change in $Step_{3-5}$ can be explained by either an *Old-External-Worsen* or a *New-External-Improve*. The former means the old route to the prefix worsens due to an external factor (*e.g.*, a prefix withdrawal, a longer AS path, a higher origin type, or a higher MED value). The latter implies the new route to the prefix improves due to a prefix announcement, a shorter AS path, a lower origin type, or a lower MED value.
- A change in $Step_7$ can be explained by an *Old-Internal-Increase* or a *New-Internal-Decrease*. The former implies the cost of the old internal path increases due to a more costly PoP-level link. The latter implies a less costly new internal path.

Same ingress PoP and same egress PoP changes

- When the internal PoP path changes, it can be explained by an *Old-Internal-Increase* or a *New-Internal-Decrease*.
- When the next hop AS changes, it can be explained by an *Old-Peering-Down*, a *New-Peering-Up*, an *Old-External-Worsen*, or a *New-External-Improve*.
- When the AS path changes but no next hop AS changes, it can be due to an *External-AS-Change*, which is not directly related to the target ISP.

Using the above rules, we can map each event to a set of possible causes. By aggregating events that occur closely in time (identified between the same pair of consecutive routing state), we construct *a bipartite graph*, called *evidence graph*, as shown in Figure 3. There are two types of nodes in an evidence graph: cause nodes at the bottom and event nodes at the top. An edge between a cause node and an event node indicates the event can be explained by the cause. An evidence graph encapsulates the relationship between all the possible causes and their supporting evidence (events).

Conflicts may exist between causes and measurement traces due to noise and errors. For instance, an *Old-Peering-Down* will conflict with a new trace which traverses the peering that is inferred to be down. Conflicts stem from two major sources: i) the subtle timing difference when traceroute probes from different end systems traverse the same peering or measure the same prefix; and ii) errors in the IP-to-AS or IP-to-PoP mappings.

A measurement trace will never conflict with an *Old-Internal-Increase* or a *New-Internal-Decrease* because a cost change on a PoP-level link may not prevent a path from using the link. However, a measurement trace may conflict with each of the remaining six causes:

- *Old-Peering-Down*: a new path still uses a peering that is inferred to be down.
- *New-Peering-Up*: an old path already used a peering that is inferred to be up.
- *Old-Lpref-Decrease*: a new path still uses an egress that is inferred to have a lower $Lpref$ even when there are other egresses with a higher $Lpref$.
- *New-Lpref-Increase*: an old path already used an egress that is inferred to have a higher $Lpref$ (therefore used to have a lower $Lpref$) even when there were other egresses with a higher $Lpref$.
- *Old-External-Worsen*: a new path still uses an old route to a prefix even when it is worse than a new route to the same prefix, or an old path already used a new route to a prefix even when the old route to the same prefix was better.
- *New-External-Improve*: a new path still uses an old route to a prefix even when a new route to the same prefix is better, or an old path already used a new route to a prefix even when it was worse than an old route to the same prefix.

We encapsulate the relationship among all the possible causes and their conflicting measurement traces using a *conflict graph*, as shown in Figure 3. Similar to an evidence graph, it has two types of nodes: cause nodes at the bottom and measurement nodes at the top. An edge between a cause node and a measurement node indicates a conflict between the cause and the measurement trace. For each evidence graph, we construct a conflict graph accordingly by inspecting all the measurement traces in the same pair of consecutive routing state. When a measurement trace conflicts with some causes in the evidence graph, we insert a measurement node and the corresponding edges into the conflict graph.

5.2 Inference algorithm

We now present our inference algorithm that uses the evidence graph and the conflict graph to infer likely causes. Our inference is guided by two rules: i) Simplest explanation is most likely to be true. We try to find the minimum set of causes that can explain all the evidence (events). ii) We should take into account the noise and errors in our measurement by minimizing conflicts between inferred causes and measurement traces.

We use a greedy algorithm to infer causes. In each iteration, it selects a cause from the evidence graph with the maximum value of $(E-\alpha C)$, where E is the number of supporting events and C is the number of conflicting traces (computed from the conflict graph). Intuitively, it selects a cause that explains many events but raises few conflicts. It then removes the events that have been explained by the cause from the evidence graph before entering the next iteration. This process continues until

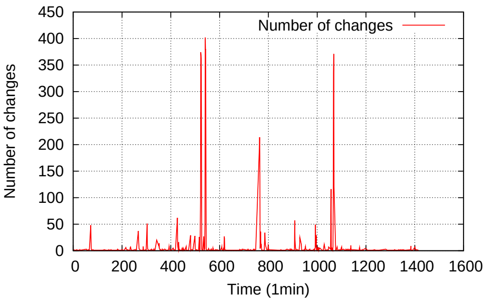


Figure 4: Number of detected changes on Sep. 25, 2007

AS Name ASN (Tier)	Periods	# of Src	# of PoPs	# of Probes	Probe Gap
AT&T 7018 (1)	3/23-4/9	230	111	61453	18.3 min
Verio 2914 (1)	4/10-4/22 9/13-9/22	218	46	81024	19.3 min
Deutsche Tele -kom 3320 (2)	4/23-5/22	149	64	27958	17.5 min
Savvis 3561 (1)	5/23-6/24	178	39	40989	17.4 min
Abilene 11537 (3)	9/23-9/30 2/3-2/17	113	11	51037	18.4 min

Table 2: Summary of data collection

all the events have been explained.

The parameter α allows us to tune the relative weight between evidence and conflicts. A larger α makes our algorithm more aggressive in avoiding conflicts. Currently, we set $\alpha = 1$ in our experiments. However, we find our results are not very sensitive to the choice of α between 0.1 and 10. This is likely due to the fact that the number of evidence significantly outweighs the number of conflicts for most causes (see §7).

Given that the inputs to our algorithm (the evidence graph and the conflict graph) are limited by the coverage of our system and measurement noise and errors, it may report incorrect causes or miss true causes. To highlight the reliability of inferred causes, we introduce a notion of *inference confidence* for each cause as $E - \alpha C$, where E and C have the same meaning as in the above. Intuitively, causes with a higher inference confidence, *i.e.*, with more evidence but fewer conflicts, are more reliable. We will demonstrate how inference confidence affects the accuracy in §7.

6 Results of Event Identification and Classification

In this section, we present the results of event identification and classification using our framework over a period of 132 days for five backbone ISPs. We validate the identified routing events using BGP data from many vantage points at the end of the section.

The summary of data collection is shown in Table 2. We study three Tier-1 ASes, one Tier-2 AS, and one Tier-3 AS. As the first step, we study one AS at a time. We plan to study multiple ASes simultaneously in the future to better diagnose routing events at a global scale. Table 2 shows the number of probing source hosts used and the number of PoPs covered. Note that there is some variability across the number of source hosts used as not all hosts are useful for improving the coverage of PoP-prefix pairs. This provides room for probing multiple ASes at the same time. We verified our PoP coverage completeness using the data from Rocketfuel [11] and router configuration files from the Abilene network. Table 2 also shows the average number of probes to acquire the routing state of a target ISP. Depending on the ISP, each source host has to probe between 187 and 371 destinations on average. As expected, our system can refresh the routing state roughly every *eighteen* minutes.

Before delving into details, we first use one example to illustrate that our system is able to detect significant network disruptions that generate a large number of routing events. Figure 4 shows the number of routing events detected using our system for Abilene over time on Sep. 25, 2007. It is clear that the routing event occurrence is not evenly distributed. We do observe a few spikes across the day. The constant background noise is often due to routing events that only affect individual prefixes. The spike around $540min$ is an internal disruption causing the egress PoP to shift from Washington DC to New York, affecting 782 source-destination pairs. The next spike around $765min$ is due to one neighbor AS2637 withdrawing routes to 112 prefixes from the Atlanta PoP. The last spike around $1069min$ is due to a peering link failure, resulting in the next hop AS in Washington DC changing from AS1299 to AS20965. All these causes have been confirmed using the BGP and the Syslog data of Abilene.

6.1 Data cleaning process

As mentioned in §4, we first need to remove the noise in our dataset. Table 3 shows the overall statistics of average daily traces removed due to various reasons. It is expected that a relatively small percentage (0.75%) of traces are ignored due to contiguous "*" hops and temporary loops. We also found that 0.025% of the traces contain persistent IP or AS loops usually occurring close to the destination, which confirms observations from a previous study [16].

Note that 3.2% of the traces are discarded due to not traversing the target ISP, as we cannot distinguish between the target ISP losing reachability or any of the preceding ISPs changing routes. One noteworthy observation is that 35% of the traces stop before entering the destination network. Most of these networks appear persistently unreachable over time, likely due to ICMP filtering at the edges between a provider and its customers. We still use these traces as they can help detect routing

	IP loop	PoP loop	AS loop	IP star	PoP star	AS star	No targetAS	Persistent IP loop	Persistent AS loop
Removed traces	12643	9934	1053	14055	5836	9573	2466927	1738	445
(percentage)	0.18%	0.14%	0.015%	0.2%	0.08%	0.13%	3.2%	0.02%	0.005%

Table 3: Statistics of data cleaning: avg number of removed traces per day for each type of anomalous traceroute.

Target AS	Total events (% all traces)	Ingress same Egress change	Ingress same, Egress same		Ingress change
			internal pop path	external AS path	
7018	277435	33325	213562 , 76.9%		30548
	0.35%	12.1%	51%	35%	11%
2914	415778	113507	261525 , 62.9%		40746
	0.31%	27.3%	48%	19%	9.8%
3320	437125	21419	384233 , 87.9%		31473
	0.66%	4.9%	8.5%	80.7%	7.2%
3561	311886	34307	233915 , 75%		43664
	0.35%	11%	45%	31%	14%
11537	145034	19776	99309 , 68%		25949
	0.24%	13.6%	37%	40%	17%

Table 4: Statistics of event classification

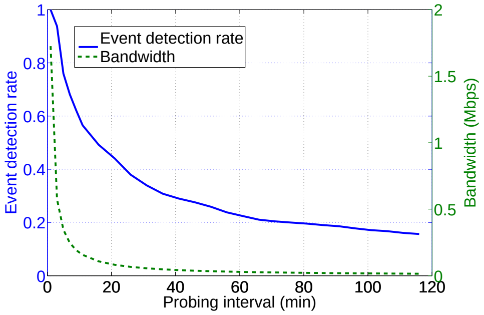


Figure 5: Impact of probing interval on detection rate and bandwidth.

changes in the partial path before filtering.

6.2 Event identification and classification

We first classify routing events according to the ingress and egress PoP changes. Table 4 shows the statistics of event classification for each ISP during our study. Only a very small fraction of the traces contain routing changes. Among these changes, a small percentage (7.2% - 17%) is found to be ingress PoP changes, because most of the probing sources enter the target AS from an ingress PoP near its geographic location. The majority (62.9% - 87.9%) of the events are in the category of both ingress and egress staying the same. This category contains either internal PoP-level path changes and/or the external AS path changes. The remaining events (4.9% - 27.3%) involve egress PoP changes. Some of these events may impose significant impact on the target ISP as a large amount of traffic to many prefixes shifts internal paths simultaneously.

Abilene, the educational backbone network, was expected to be stable due to its simple topology. Surprisingly, we found that it has a larger fraction of ingress changes. This is observed mainly from three source hosts, switching their ingress PoP to various destinations. Two of them are universities in Oregon, with access links to Abilene in both Seattle and Los Angeles. The other one is a university in Florida, which has access links in both Atlanta and Kansas City. We confirm this via the Abilene border routers' configuration files. We believe this could be due to load-balancing or traffic engineering near the sources.

6.3 Validation with BGP data

Using public BGP feeds from RouteViews, RIPE and Abilene, in addition to 29 BGP feeds from a Tier-1 ISP, we validate our results in two aspects: the *destination prefix coverage* and the *routing event detection rate*. We omit AS3320 here due to lack of access to its BGP data.

To evaluate the destination prefix coverage of our dataset, we map the destination IP to the longest prefix using the latest routing table of each AS. Then by comparing the set of probed prefixes with all the prefixes in the default-free routing table of each target AS, we compute the coverage, as shown in the second column of Table 5. Although the coverage is only between 6% to 18%, our traces cover all the known distinct PoP-level links within each target AS (compared to the Rocketfuel data [11]), suggesting that we can detect significant routing changes originated inside the target AS.

We use the following methodology for validating changes detected using BGP data. For the five ASes we studied, we only have BGP feeds for four ASes. For each of them, we first identify the corresponding PoP where the BGP feed comes from. Because different PoPs in an AS usually experience different routing changes, we compare BGP-observed changes with traceroute-observed changes only when our traces traverse the PoP where the BGP feed comes from. The third column of Table 5 shows the ratio of the probed destination prefixes that traverse the PoP of the BGP feed relative to the total number of prefixes in a default-free routing table.

The subset of destinations which can be used for comparison varies across ASes due to the different number of available BGP feeds. We focus on examining for any BGP-observed routing change of this subset of destinations, whether we also detect it using our traces. Moreover, we only account for BGP routing changes with either AS path changes or next hop AS changes, which can

Target AS	Dst. prefix coverage	Dst. prefix traversing PoPs with BGP feeds	Detected events (AS change, nexthop change)	Missed events (short duration, filtering, others)
7018	34145 (15%)	3414 (1.5%)	64714, 11% (10.3%, 3.2%)	89% (75%, 13%, 1%)
2914	40881 (18.6%)	40039 (18.1%)	73689, 23% (19.1%, 8.6%)	77% (73%, 4%, 0%)
3561	17317 (7.8%)	2317 (1.1%)	55692, 6% (5.8%, 0.5%)	94% (80%, 9%, 5%)
11537	13789 (6%)	13789 (6%)	66706, 21% (17.3%, 5.8%)	79% (61%, 15%, 3%)

Table 5: Validation with BGP data for dst. prefix coverage and event detection rate.

be detected via traceroute. By comparing the two sets, we calculate the fraction of changes our system can detect, as shown in the fourth column of Table 5. This rate varies between 6% to 23%. Note that we can also detect many internal PoP path changes which are not observed in the BGP data (thus not included in this table).

Changes missed by our system are due to two main reasons. First, the routing changes last too short to be detected by two consecutive probes, accounting for the majority of the missed routing events. As explained in §3, we do not focus on these short-lived routing events. We are able to detect most events with duration larger than 20 minutes (probing interval). Given that we cannot detect routing events that last shorter than the probing interval, we may increase the event detection rate by reducing the probing interval. Figure 5 illustrates how the probing interval affects the event detection rate and probing bandwidth. When the probing interval is 10 minutes, we can detect 60% of the events while using roughly 0.2 Mbps bandwidth.

Second, because traceroute may be incomplete due to packet filtering, certain changes cannot be detected as the changing path segment is invisible from our probes. Most filtering happens in the path segment after the next hop AS and close to the destination AS. Since we only use the next hop AS information for event correlation, missing these changes does not have any impact on our inference results.

Only a small fraction (up to 5%) of the missed changes are due to other factors, *e.g.*, inaccurate IP-to-AS mappings or mismatched forward paths compared to the BGP data. In summary, our system is able to capture most routing changes to the probed destinations that are useful for event correlation and inference.

7 Results of Event Correlation and Inference

In this section, we first present the results of our inference algorithm. Then we validate our system in three ways: comparing with the BGP feed based inference using BGP data from a Tier-1 ISP, comparing with both BGP data and Syslog data from the Abilene network, and comparing with disruptions reported from the NANOG email list [10].

7.1 Result summary

Our inference algorithm takes the set of identified events and automatically clusters them based on their causes. Table 6 shows both the total number and the relative percentage for each type of causes inferred for each ISP. We observe that different ISPs can have a non-negligible difference in the cause distribution. For example, for the first three ISPs, the largest fraction of events are caused by *External-AS-Change*. In contrast, Abilene (AS11537) has more events caused by *Old-External-Worsen* and *New-External-Improve*. This is mainly caused by its five neighbor ASes. The most dominant one is the neighbor AS20965 peering in New York which switches routes to around 390 destinations frequently over time.

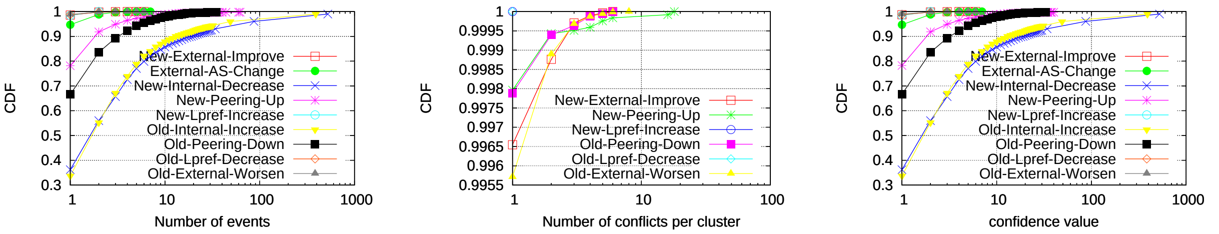
We study the effectiveness of our inference algorithm in clustering related events together in Figure 6(a). A cluster is defined to be the set of events explained by a single cause. The figure shows the CDF of the number of events per cluster over the entire period for five ASes. While most of them have less than ten events per cluster, there are some clusters with many events, indicating significant routing disruptions. *New-Internal-Decrease*, *Old-Internal-Increase*, *Old-Peering-Down*, and *New-Peering-Up* have relatively larger clusters than others, confirming previous findings that hot-potato changes and peering session up/down can impose significant impact [18]. Other types of causes have much smaller clusters, because they usually only affect individual prefixes.

Another metric to evaluate the accuracy of inferred cause is based on the number of conflicts introduced by the cause, as shown in Figure 6(b). According to §5, only six types of causes may have conflicts. Overall, the number of conflicts per cluster is small compared to the number of events per cluster, indicating that the inconsistencies in our traces introduced by incorrect mappings or differences in probing time are rare.

We use the confidence metric introduced in the previous section to assess the likelihood of causes. Figure 6(c) shows that different types of causes have different distributions of confidence value. For example, *Old-External-Worsen*, *New-External-Improve*, *Old-Lpref-Decrease*, and *New-Lpref-Increase* generally have much lower confidence values as they affect only indi-

Target AS	Old-Int. -Increase	New-Int. -Decrease	Old Peering Down	New Peer -ing Up	Old-Ext. -Worsen	New-Ext. -Improve	Old-Lpref -Decrease	New-Lpref -Increase	Ext. AS Change
7018	5223, 4.5%	3843, 3%	5677, 5%	4955, 4.3%	18142, 16%	20961, 18%	302, 0.2%	397, 0.3%	55216, 48%
2914	10366, 5%	8135, 4%	6666, 4%	7024, 3.7%	38748, 20%	49075, 26%	124, 0.1%	164, 0.1%	69190, 36%
3320	1622, 0.5%	954,0.2%	20751, 5%	10204, 3%	80385, 21%	81761, 21%	751, 0.2%	1002,0.2%	185683, 48%
3561	4410, 3.6%	4007, 3%	6017, 5%	7667, 6.3%	23232, 19%	45495, 37%	85, 0.1%	105, 0.1%	30540, 25%
11537	2161, 1.8%	1632, 1%	2771, 2%	1401, 1.1%	44516, 37%	43375, 36%	112, 0.1%	104, 0.1%	9589, 8%

Table 6: Statistics of cause inference.



(a) # of events per cluster

(b) # of conflicts per cluster

(c) Confidence value per cluster

Figure 6: Events, conflicts and confidence value distribution per cluster.

vidual prefixes. Thus we need to set appropriate thresholds to filter out different types of causes with low confidence. Throughout the rest of this section, we use a confidence value of 30 for reporting hot-potato changes (*Old-Internal-Increase* and *New-Internal-Decrease*) and 150 for reporting peering session changes (*Old-Peering-Down* and *New-Peering-Up*). A lower confidence value increases the likelihood of false positives, *e.g.*, misinterpreting multiple simultaneous prefix withdrawals from a peering as an *Old-Peering-Down*. These two confidence values filter out 92% of the hot-potato changes and 99% of the peering session changes inferred without using any thresholds. We will evaluate the impact of the confidence value on our inference accuracy in the next section. We do not set any threshold for other types of causes since most of them have only one event in each cluster.

7.2 Validation with BGP-based inference for a Tier-1 ISP

Most previous work on diagnosing routing disruptions relies on BGP data. The closest one to ours is by Wu *et al.* [6] using BGP updates from all the border routers to peers to identify important routing disruptions. To directly compare with their approach, we implemented their algorithm, called *Wu* for convenience. We collected data via eBGP sessions to 29 border routers in a Tier-1 ISP. Note that *Wu* requires BGP data from all the border routers and focuses on peer routes only. Given the lack of access to such complete data, causes reported by *Wu* on our data may be inaccurate accounting for possible mismatches.

We briefly summarize *Wu*'s algorithm and our comparison methodology. *Wu* first groups a routing event from one border router's perspective into five types: no change, internal path change (using iBGP routes with nexthop change), loss of egress point (changing from eBGP to iBGP route), gain of egress point (changing from iBGP to eBGP route), and external path change (both using eBGP route with nexthop change). This step is accurate even with incomplete data. By correlating events from individual routers, *Wu* generates a vector of events for each destination prefix to summarize how the route for each prefix has changed. The types of changes include: *transient disruption*, *internal disruption* (all routers experience internal path change), *single external disruption* (only one router has either loss/gain of egress or external change), *multiple external disruption* (multiple routers have either loss/gain of egress or external changes), and *loss/gain of reachability* (every router experiences loss/gain of egress). This step may introduce inaccuracy due to data incompleteness. Note that incomplete data set can only cause *Wu* to falsely categorize external events into internal events.

We first validate our event classification results by comparing with *Wu*'s vector change report. We map each of our events (per source-destination based routing change) to the corresponding event in *Wu*, the prefix of which covers our destination. Each event is associated with one cause from our algorithm and one vector change type in *Wu*. Note that the set of causes and the set of vector change types do not have direct one-to-one mapping. To perform comparison, we combine our causes into two big categories:

Internal includes *New-Internal-Decrease*, *Old-Internal-Increase*, *Old-Lpref-Decrease*, *New-Lpref-Increase*, which should match *Wu's internal disruption*.

External includes *Old-External-Worsen*, *New-External-Improve*, *Old-Peering-Down*, *New-Peering-*

Root cause	Internal disruption	Single external	Multiple external	Loss/gain of reachability
Internal	***34914 (76.9%)***	5947 (13.1%)	4494 (9.9%)	10 (0.02%)
External	16344 (24.2%)	***44948 (65.9%)***	***6538 (9.6%)***	391 (0.6%)

Table 7: Event based validation: with a Tier-1 ISP's BGP data over 21 days.

Target AS	Hot potato			Session reset		
	Wu	Our	Both	*Wu*	Our	Both
Tier-1 ISP	147	185	101 68%,55%	9	15	6 66%,40%
Abilene (11537)	79	88	60 76%,68%	7	11	7 100%,63%

Table 8: Validation for two important clusters ($confidence_{hotPotato}$=30, $confidence_{session}$=150)

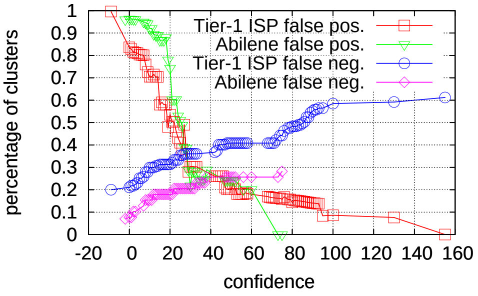


Figure 7: Inference accuracy for hot-potato changes – a common type of routing disruption.

Up, which should match *Wu's single/multiple external disruption*.

These two aggregate categories are of interest because our main goal is to distinguish internal disruptions from external ones. The cause *External-AS-Change* does not have any corresponding type in *Wu*, which is thus omitted from comparison. Similarly, we omit our Same-Ingress-Same-Egress type of events with only internal PoP path changes, as it is not considered by *Wu*.

As shown in Table 7, each column is the type of vector change in *Wu*, while each row shows our aggregate categories. For each routing event, we identify the type y inferred from Wu as well as the category x inferred by our system. By comparing them, we generate the percentage in the table row x column y which is the fraction of events in our aggregate category x that is categorized as type y in *Wu*. The cell with bold italic font means valid matches. 76.9% of our internal events match *Wu's internal disruption*, while 75.5% of our external events match *Wu's single/multiple external disruption*. While the match rate of around 75% is not very high, we believe our end-system based approach shows promise in inferring routing disruptions and the rate can be further improved with more vantage points.

The third step in *Wu* is to group together event vectors of different destinations belonging to the same type and transition trend. There are two types of clusters reported in the third step: hot-potato changes and peering session resets. For each of the causes reported by us, we examine if it is also reported by *Wu*. To be more specific, for each *New-Internal-Decrease* and *Old-Internal-Increase*, we search for the corresponding hot-potato changes reported within that probing interval. Each *Old-Peering-Down* and *New-Peering-Up* is mapped to *Wu's peering session reset* in the same probing interval associated with the same egress and neighbor AS.

The comparison for these two important clusters is shown in Table 8. We use the confidence value of 30 for hot-potato changes and 150 for session resets based on their distinct confidence distributions shown in the previous section. The two algorithms reported 101 common hot-potato changes and 6 common session resets. Given that our system does not rely on any ISP proprietary data, it is quite encouraging that we can correctly diagnose a reasonably large fraction of significant routing disruptions (68% of hot-potato changes and 66% of session resets).

We study the impact of confidence value on our inference accuracy of hot-potato changes in Figure 7. As expected, with larger confidence values, the false positive rate decreases while the false negative rate increases. With a confidence threshold of 30, we attain a balance between false positives (45%) and false negatives (32%). Similarly, for session reset, the false positive and false negative rates are 60% and 34% respectively with a confidence value threshold of 150.

7.3 Validation with BGP-based inference and Syslog analysis for Abilene

We also validate our inference results with *Wu*'s algorithm executed on the BGP data from all 11 border routers of the Abilene network [19]. This provides a more complete view of routing changes for the entire network compared to the Tier-1 ISP case. Besides BGP data, router Syslog messages are also available [19] from all the Abilene border routers. Syslog reports error messages such as link down events due to hardware failure or maintenance. We can thus validate inferred link up/down causes directly using Syslog messages.

Table 9 compares the routing event inference between *Wu* and our system. The match rate for Abilene is higher compared to the Tier-1 ISP case, due to the improved accuracy of *Wu* given full visibility. 7.3% of the internal disruptions are mis-classified as external disruptions, most likely due to the limited coverage of our system. When an internal path is traversed only a few times, it

Cause	Internal disruption	Single external	Multiple external	Loss/gain of reachability
Internal	***4463*** ***(85%)***	1059 (7.2%)	837 (8%)	2% (0.01%)
External	2929 (7.3%)	***21642*** ***(86.4%)***	***2355*** ***(6.2%)***	79 (0.1%)

Table 9: Event based validation: with Abilene's BGP data over 21 days.

is less likely to be selected by our greedy algorithm as the cause of routing events. This problem could be mitigated by using more vantage points or increasing the confidence level threshold.

The comparison for the two important clusters is shown in Table 8. From the Abilene Syslog, the seven session resets were caused by peering link down events which lasted for more than fifteen minutes, possibly due to maintenance. Overall, we correctly inferred 76% of the hot-potato changes and 100% of the session resets. The false positive rates are 32% for hot-potato changes and 37% for session resets respectively.

7.4 Validation with NANOG mailing list

Given that operators today often use the NANOG (North American Network Operators Group) mailing list [10] to troubleshoot network problems, we study the archives of the mailing list messages over the time period of our study. All together we analyzed 2,694 emails using keyword searches and identified six significant routing disruptions with details described below. One interesting observation is that even though we did not directly probe the problematic ASes described in the emails, we are still able to identify the impact and infer the causes relative to the target ASes for the following four events due to their wide-spread impact:

1. Apr. 25, 2007, between 19:40 to 21:20 EDT, NANOG reported a Tier-1 ISP Cogent (AS174) experienced serious problem on its peering links causing many route withdrawals. The target AS during this time was AS3320. Our system observed increased number of routing events: 120 detected events were clustered into 96 causes of *External-AS-Change*, affecting 7 sources and 118 destinations. 87 of the events were associated with 42 destinations which were Cogent's customers. They all switched from routes traversing Cogent. Significant delay increase was also observed.

2. May 21, 2007, around 21:50 EDT, NANOG reported a backbone link fiber cut between Portland and Seattle in the Level3 network (AS3356), resulting in reachability problems from Level3's customers. The target AS at that time was also AS3320. Our system detected 45 events clustered into 36 causes of *Old-External-Worsen*, affecting 5 probing sources and 12 destinations. They all switched from routes traversing Level3 to those traversing AS3491 in the Seattle PoP.

3. Jun. 14, 2007, NANOG reported a core router outage around 6am EDT in the Qwest network (AS209), affecting the performance of several networks and their customers. The target AS studied at the time was AS3561. Our system reported 24 events clustered into 23 causes of *External-AS-Change* switching from paths through AS209 to those traversing AT&T (AS7018) around the outage time, affecting 6 probing sources and 24 destinations.

4. Sep. 19, 2007, 13:00 EDT, NANOG reported that 25 routers in the Broadwing network (AS6395) had a misconfiguration resulting in BGP session removal. It caused multiple single-homed customers disconnected from the Internet. Immediately after that, our system detected 81 events clustered into 64 causes of *Old-External-Worsen*, for 76 destinations from 10 sources. The target AS, AS2914, switched from the old routes traversing Level3 (AS3356) and Broadwing to new routes traversing other peers, *e.g.,* AS209 and AS7018.

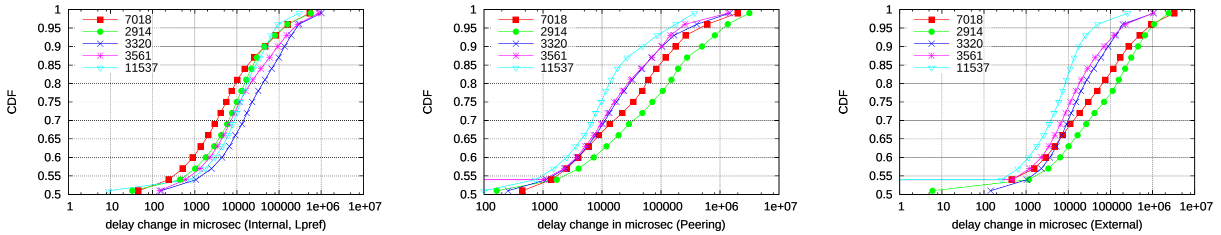
We missed two NANOG-reported events related to routing and performance disruptions during our study. The first was on May 16, 2007, from 13:10 to 14:20 EDT, related to a hardware problem on the peering link between AT&T and Broadwing in Dallas. Our system did not capture any routing changes during this time period at that location. The second event was on May 30, 2007, around 13:00 EDT, related to significant performance degradation, along with temporary loss of reachability from Sprint in the Pittsburgh area, as confirmed from Sprint. The target AS probed was AS3561. Although our system did not report routing changes related to Sprint, it did observe abnormal incomplete traces from PlanetLab hosts in Pittsburgh.

To summarize, our system may miss some localized disruptions due to limited coverage. However, it is able to capture disruptions with global impact even when they are not directly caused by the target AS being probed.

8 Performance Impact Analysis

Routing events are known to introduce disruption to network path performance. Unlike the past work that relies on routing feeds to diagnose routing events, end-host probing used in our system enables us to understand the impact of routing events on path performance. In this section, we study to what extent end-to-end latency is affected by different types of routing events and its variation cross different ISPs.

Figure 8 illustrates the latency change for different type routing events in AS7018. For clarity, we only show five types of events: *Internal* (*Old-Internal-Increase*, *New-Internal-Decrease*), *Peering* (*Old-Peering-Down*,



(a) Internal and local pref changes

(b) Peering changes

(c) External changes

Figure 9: Delay change distribution across ISPs

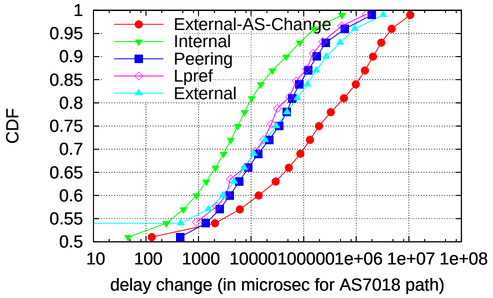


Figure 8: Delay change distribution of each category for AS7018.

New-Peering-Up), *Lpref* (*Old-Lpref-Decrease*, *New-Lpref-Increase*), *External* (*Old-External-Worsen*, *New-External-Improve*), and *External-AS-Change*. Because we use log scale on the y-axis, the graph does not show the cases where latency change is negative. Given that almost all the curves start from 0.5, it implies latency has the same likelihood to improve or worsen after these events. A noteworthy observation is external events (*External-AS-Change*, *External*, and *Peering*) have much more severe impact, suggesting that AT&T's network is engineered well internally. We observe similar patterns for the other ISPs studied.

Figure 9 illustrates how the latency change induced by the same event type varies across different ISPs. We omit *External-AS-Change* here because this type is not directly related to a target ISP. Figure 9(a) shows little difference among the five target ISPs in terms of latency change caused by internal events, as most changes are relatively small. Turning to Figure 9(b) and (c), the difference between the ISPs becomes much more noticeable. AS11537 appears most resilient to external events in terms of latency deterioration while AS2914 appears worst. The relative difference between the ISPs is consistent in both graphs, suggesting that customers sensitive to performance disruptions should take great care in selecting the appropriate ISP providers.

9 System Evaluation

In this section, we show that our system imposes a small amount of memory and CPU overhead to perform event identification, classification, and inference. We evaluate our system on a commodity server with eight 3.2GHz Xeon processors and 4 GB memory running Linux 2.6.20 SMP.

The memory usage of our system is composed of: i) the two most recent routing state of the target ISP extracted from the traces; and ii) the evidence and the conflict graphs constructed from the two routing state (see §3). The former is relatively static over time since the overall topology and routing of a target ISP do not change frequently. The latter is more dynamic and depends on the number of detected routing events. Throughout our evaluation period, the former is dominant because the number of traces outweighs the number of routing events. The total memory footprint of our system stays under 40 MB. We also evaluate whether our system can keep up with the continually incoming routing state. We find the processing time of two recent routing state never exceeds one eighth of the data collection time between the two routing state. This suggests our system can operate in real time to quickly detect and raise alerts on significant routing disruptions.

10 Related Work

Much work has been proposed to use end-host based probing to identify various network properties. For example, Rocketfuel [11] discovers ISP topologies by launching traceroute from a set of hosts in an intelligent manner to ensure scalability and coverage. iPlane [20] estimates the Internet path performance using traceroutes and prediction techniques. There exist many other research measurement infrastructures [21, 22, 23, 24, 25] for measuring network distance with performance metrics such as latency and bandwidth. Another example is PlanetSeer [26] which uses active probes to identify performance anomalies for distributed applications. The key difference from these measurement efforts is that our work focuses on using collaborative traceroute probes to diagnose routing changes associated with large networks.

The closest related work on identifying routing disruptions is that by Wu *et al.* [6]. Using BGP data from multiple border routers in a single ISP, their system identifies significant BGP routing changes impacting large amount of traffic. A follow-up work by Huang *et al.* [27] performs multivariate analysis using BGP data from all routers within a large network combined with router configurations to diagnose network disruptions. In contrast, we do not rely on such proprietary BGP data, and we can apply our system to diagnose routing changes for multiple networks. Another closely related work is the Hubble system [28] which attempts to identify reachability problems using end-system based probing. In contrast to their work, we attempt to both identify routing events and infer their causes relative to the target AS. There are also several projects on identifying the location and causes of routing changes by analyzing BGP data from multiple ASes [5, 9]. However, it is difficult to have complete visibility due to a limited number of BGP monitors. Note that our system is not restricted by the deployment of route monitors and can thus be widely deployed.

11 Conclusion

In this paper we have presented the first system to accurately and scalably diagnose routing disruptions purely from end systems without access to any sensitive data such as BGP feeds or router configurations from ISP networks. Using a simple greedy algorithm on two bipartite graphs representing observed routing events, possible causes, and the constraints between them, our system effectively infers the most likely causes for routing events detected through light-weight traceroute probes. We comprehensively validate the accuracy of our results by comparing with an existing ISP-centric method, publicly-available router configurations, and network operators' mailing list. We believe our work is an important step to empowering customers and ISPs for attaining better accountability on today's Internet.

References

[1] F. Wang, Z. M. Mao, J. Wang, L. Gao, and R. Bush, "A Measurement Study on the Impact of Routing Events on End-to-End Internet Path Performance," in *Proc. ACM SIGCOMM*, 2006.

[2] Y. Zhang, Z. M. Mao, and J. Wang, "A Framework for Measuring and Predicting the Impact of Routing Changes," in *Proc. IEEE INFOCOM*, 2007.

[3] H. Pucha, Y. Zhang, Z. M. Mao, and Y. C. Hu, "Understanding network delay changes caused by routing events," in *Proc. ACM SIGMETRICS*, 2007.

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

[5] A. Feldmann, O. Maennel, Z. M. Mao, A. Berger, and B. Maggs, "Locating Internet Routing Instabilities," in *Proceedings of ACM SIGCOMM*, 2004.

[6] J. Wu, Z. M. Mao, J. Rexford, and J. Wang, "Finding a needle in a haystack: Pinpointing significant BGP routing changes in an IP network," in *Proc. Symposium on Networked Systems Design and Implementation*, 2005.

[7] "University of Oregon Route Views Archive Project." `www.routeviews.org`.

[8] "Ripe NCC." `http://www.ripe.net/ripencc/pub-services/np/ris/`.

[9] R. Teixeira and J. Rexford, "A measurement framework for pinpointing routing changes," in *NetT '04: Proceedings of the ACM SIGCOMM workshop on Network troubleshooting*, 2004.

[10] "NANOG Mailing List Information." `http://www.nanog.org/mailinglist.html`.

[11] N. Spring, R. Mahajan, D. Wetherall, and T. Anderson, "Measuring isp topologies with rocketfuel," *IEEE/ACM Trans. Netw.*, vol. 12, no. 1, pp. 2–16, 2004.

[12] R. Mahajan, M. Zhang, L. Poole, and V. Pai, "Uncovering Performance Differences in Backbone ISPs with Netdiff," in *Proceeding of NSDI*, 2008.

[13] N. Spring, D. Wetherall, and T. Anderson, "Scriptroute: A Public Internet Measurement Facility," in *Proceedings of USENIX Symposium on Internet Technologies and Systems (USITS)*, 2003.

[14] Z. M. Mao, J. Rexford, J. Wang, and R. Katz, " Towards an Accurate AS-Level Traceroute Tool ," in *Proc. ACM SIGCOMM*, 2003.

[15] C. Labovitz, A. Ahuja, A. Bose, and F. Jahanian, "Delayed internet routing convergence," in *Proc. ACM SIGCOMM*, 2000.

[16] J. Xia, L. Gao, and T. Fei, "Flooding Attacks by Exploiting Persistent Forwarding Loops," in *Proc. ACM SIGCOMM IMC*, 2005.

[17] R. Kompella, J. Yates, A. Greenberg, and A. C. Snoeren, "IP fault localization via risk modeling," in *Proc. Symposium on Networked Systems Design and Implementation*, 2005.

[18] R. Teixeira, A. Shaikh, T. Griffin, and J. Rexford, "Dynamics of hot-potato routing in ip networks," *SIGMETRICS Perform. Eval. Rev.*, vol. 32, no. 1, pp. 307–319, 2004.

[19] "Internet2 Network NOC - Research Data." `http://www.abilene.iu.edu/i2network/research-data.html`.

[20] H. V. Madhyastha, T. Isdal, M. Piatek, C. Dixon, T. Anderson, A. Krishnamurthy, and A. Venkataramani, "iPlane: An Information Plane for Distributed Services," in *Proc. Symposium on Operating Systems Design and Implementation*, 2006.

[21] P. Francis, S. Jamin, V. Paxson, L. Zhang, D. Gryniewicz, and Y. Jin, "An Architecture for a Global Internet Host Distance Estimation Service," in *Proceedings of IEEE INFOCOM*, 1999.

[22] T. S. E. Ng and H. Zhang, "Predicting Internet Network Distance with Coordinates-Based Approaches," in *Proceedings of IEEE INFOCOM*, June 2002.

[23] R. Govindan and H. Tangmunarunkit, "Heuristics for Internet Map Discovery," in *Proc. of IEEE INFOCOM*, (Tel Aviv, Israel), pp. 1371–1380, March 2000.

[24] F. Dabek, R. Cox, F. Kaashoek, and R. Morris, "Vivaldi: A Decentralized Network Coordinate System," in *Proceedings of ACM SIGCOMM*, August 2004.

[25] M. Costa, M. Castro, A. Rowstron, and P. Key, "PIC: Practical Internet Coordinates for Distance Estimation," in *Proceedings of IEEE ICDCS*, March 2004.

[26] M. Zhang, C. Zhang, V. Pai, L. Peterson, and R. Wang, "PlanetSeer: Internet Path Failure Monitoring and Characterization in Wide-Area Services.," in *Proc. Symposium on Operating Systems Design and Implementation*, 2004.

[27] Y. Huang, N. Feamster, A. Lakhina, and J. J. Xu, "Diagnosing network disruptions with network-wide analysis.," in *Proc. ACM SIGMETRICS*, pp. 61–72, 2007.

[28] E. Katz-Bassett, H. V. Madhyastha, J. P. John, and A. Krishnamurthy, "Studying Blackholes in the Internet with Hubble," in *Proceeding of NSDI*, 2008.