Diagnosing Network Disruptions with Network-Wide Analysis

Yiyi Huang*, Nick Feamster*, Anukool Lakhina†, Jun (Jim) Xu*
* College of Computing, Georgia Tech, † Guavus, Inc.
{yiyih,feamster}@cc.gatech.edu,anukool.lakhina@guavus.com, jx@cc.gatech.edu

ABSTRACT

To maintain high availability in the face of changing network conditions, network operators must quickly detect, identify, and react to events that cause network disruptions. One way to accomplish this goal is to monitor routing dynamics, by analyzing routing update streams collected from routers. Existing monitoring approaches typically treat streams of routing updates from different routers as independent signals, and report only the "loud" events (*i.e.*, events that involve large volume of routing messages). In this paper, we examine BGP routing data from all routers in the Abilene backbone for six months and correlate them with a catalog of all known disruptions to its nodes and links. We find that many important events are not loud enough to be detected from a single stream. Instead, they become detectable only when multiple BGP update streams are simultaneously examined. This is because routing updates exhibit *network-wide* dependencies.

This paper proposes using network-wide analysis of routing information to diagnose (*i.e.*, detect and identify) network disruptions. To detect network disruptions, we apply a multivariate analysis technique on dynamic routing information, (*i.e.*, update traffic from all the Abilene routers) and find that this technique can detect every reported disruption to nodes and links within the network with a low rate of false alarms. To identify the type of disruption, we jointly analyze both the network-wide static configuration and details in the dynamic routing updates; we find that our method can correctly explain the scenario that caused the disruption. Although much work remains to make network-wide analysis of routing data operationally practical, our results illustrate the importance and potential of such an approach.

Categories and Subject Descriptors

C.2.6 [**Computer Communication Networks**]: Internetworking; C.2.3 [**Computer Communication Networks**]: Network operations – network management

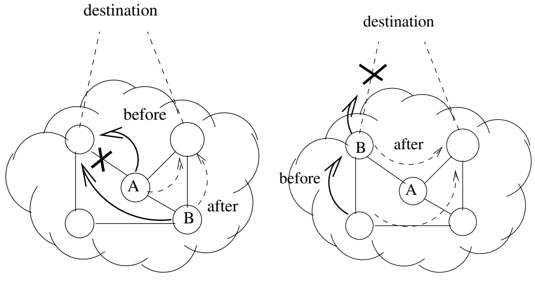
General Terms

Algorithms, Management, Reliability, Security

Keywords

anomaly detection, network management, statistical inference

SIGMETRICS'07, June 12–16, 2007, San Diego, California, USA.
Copyright 2007 ACM 978-1-59593-639-4/07/0006 ...$5.00.



(a) Internal disruption (b) External disruption

Figure 1: Both internal and external network disruptions cause correlated routing changes at groups of routers within a single network.

1. Introduction

To achieve acceptable end-to-end performance in the face of dynamic network conditions (*e.g.*, traffic shifts, link failures, security incidents, etc.), network operators must keep constant watch over the status of their networks. Network disruptions—changes in network conditions that are caused by underlying failures of routing protocols or network equipment—have a significant impact on network performance and availability. Operators today have myriad datasets (*e.g.*, NetFlow, SNMP, "syslogs") at their disposal to monitor for network disruptions, all of which have proven difficult to use for extracting actionable events from "background noise". Operators have had particular trouble using *routing data* to detect and pinpoint network disruptions, even though analyzing routing data holds promise for exposing many important network reachability failures. This missed opportunity results from the fact that routing data is voluminous, complex and noisy, which makes the mining of network disruptions challenging.

Existing approaches for inspecting routing dynamics in a single network (*e.g.*, [25, 30]) primarily analyze each routing stream without considering the dependencies across multiple routing streams that arise from the network configuration and topology. This approach leaves much room for improvement, because any information about network disruptions that exists in a single routing update stream is obscured by a massive amount of noise. Furthermore, no network model can explain the temporal relationships among updates in a single routing stream, since the updates have little (and often no) temporal dependency. As such, these techniques are unable to capture typical network conditions to recognize disruptions, and therefore rely on fixed thresholds to detect only those events that cause a large number of updates. But, as we will see in this paper, many important operational events do *not* necessarily generate a large number of updates at a single router. To detect such operational events, it is necessary to first continuously monitor and *learn* the typical routing dynamics of the network; deviations from this typical behavior indicate a routing incident worth investigating.

This paper proposes a new approach to learning typical routing dynamics by explicitly harnessing the *network-wide dependencies* that are inherent to the routing updates seen by routers in a single network. Groups of updates from different routers, when analyzed together, reflect dependencies arising from the network topology and static routing configuration: routers' locations in the network topology relative to each other, how they are connected to one another, the neighboring networks they share in common, etc. For example, Teixeira *et al.* observed that the failure of a single link inside a network may result in *multiple* routers simultaneously switching "egress routers" (*i.e.*, the router used to exit the network) [28] (Figure 1(a)); similarly, the failure of a single BGP peering session results in similar correlated disruptions across the network (Figure 1(b)). Because of these dependencies, network disruptions can appear significant when the effect of the event is viewed across all of the routers in the network, even if the number of updates seen by any single router is small.

This paper presents the first known study of network-wide correlation of routing updates in a single network, demonstrates that detection schemes should incorporate network-wide analysis of routing dynamics, and explores the extent to which multivariate analysis could expose these events. Table 1 summarizes the major findings of this paper, which presents the following contributions:

First, we study how actual, documented network disruptions are reflected in routing data. Several previous studies examine how BGP routing updates correlate with poor path performance [5, 13, 29], but these studies do not correlate BGP instability with "ground truth", known disruptions (*e.g.*, node and link failures) in an operational network. Our work examines how *known, documented network disruptions* are reflected in the BGP routing data within that network. We perform a joint analysis of documented network component failures in the Abilene network and Abilene BGP routing data for six months in 2006 and find that most network disruptions are reflected in BGP data in some way, though often not via high-volume network events.

Second, we explore how network-wide analysis can expose classes of network disruptions that are not detectable with existing techniques. After studying how known disruptions appear in BGP routing data, we explore how applying multivariate analysis techniques, which are specifically designed to analyze multiple statistical variables in parallel, could better detect these disruptions. We explore how applying a specific multivariate analysis technique, Principal Component Analysis (PCA), to routing message streams across the routers in a single network can extract network events that existing techniques would fail to detect.

Third, we present new techniques for combining analysis of routing dynamics with static configuration analysis to localize network disruptions. In addition to detecting failures, we develop algorithms to help network operators identify likely failure scenarios. Our framework helps network operators explain the source of routing faults by examining the semantics of the routing messages involved in a group of routing updates in conjunction with a model of the network, derived from static configuration analysis. This *hybrid analysis* approach is the first known framework for using a combination of routing dynamics and static routing configuration to help operators detect and isolate the source of network disruptions.

Previous work has taken on the audacious goal of Internet-wide "root cause analysis" [3, 8, 31], but all of these techniques have faced two fundamental limitations: lack of information in any single routing stream and poor knowledge of global router-level topology. In this work, we recommend revisiting the use of BGP routing data within a single network using multiple data streams, where correlations *across* streams can provide additional information about the nature of a failure, and access to network configurations can provide valuable information about the network topology (*e.g.*, the routers that have connections to a particular neighboring network). Our goal is not primarily to evaluate or optimize a specific multivariate analysis technique (*e.g.*, PCA), but rather (1) to explore the nature of how disruptions in a single network are reflected network-wide and temporally in BGP routing data, (2) to argue in general for the utility of using network-wide analysis techniques for improving detection of network disruptions and (3) to demonstrate how, once detected, network models based on static routing configurations can help operators detect and isolate the cause of these disruptions.

Finding	**Location**
Many network disruptions cause only low volumes of routing messages at any single router.	§3.2, Fig. 5
About 90% of local network disruptions are visible in BGP routing streams.	§4.1, Fig. 8
The number of updates resulting from a disruption may vary by several orders of magnitude.	§4.2, Fig. 6
About 75% of network disruptions result in near-simultaneous BGP routing messages at two or more routers.	§4.3, Fig. 8
The PCA-based subspace method detects 100% of node and link disruptions and about 60% of disruptions to peering links, with a low rate of false alarms.	§5.3, Tab. 3
The identification algorithm based on hybrid static and dynamic analysis correctly identifies 100% of node disruptions, 74% of link disruptions, and 93% of peer disruptions.	§6.3, Fig. 11

Table 1: Summary of major results.

Many hurdles must be surmounted to make our methods practical, such as (1) building a system to collect and process distributed routing streams in real time; and (2) determining the features in each signal that are most indicative of high-impact disruptions (we use number of updates, as most existing methods do, but we believe that more useful features may exist). Rather than providing the last word on analysis of routing dynamics, this paper opens a new general direction for analyzing routing data based on the following observation: *The structure and configuration of the network gives rise to dependencies across routers, and any analysis of these streams should be cognizant of these dependencies, rather than treating each routing stream as an independent signal.* In addition, we believe that our combined use of static and dynamic analysis for helping network operators identify the cause and severity of network disruptions represents an important first step in bridging the gap between static configuration analysis and monitoring of routing dynamics.

2. Background

We now present necessary background material. We first describe the general problems involved in using routing dynamics to detect and identify network disruptions. Then, we explain how changes to conditions within a single network can give rise to routing dynamics that exhibit network-wide correlations across multiple routing streams.

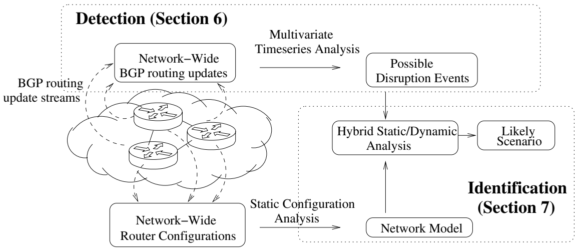


Figure 2: Overview of the approach to detection and identification of network disruptions.

2.1 Problem Overview and Approach

Diagnosis entails two complementary approaches: *proactive* techniques, which analyze the network configuration (either statically [6] or with a simulator [26]) before it is deployed; and *reactive* techniques, which observe the behavior of a running network (*e.g.*, through traffic data or routing data) and alert operators to actionable problems. Proactive analysis allows a network operator to analyze the network configurations offline and determine the effects of a set of configurations before running them on a live network [6], but it provides no mechanism for helping operators detect and identify problems in running network. To effectively detect, identify, and eradicate faults on a running network, operators must use a combination of proactive and reactive detection techniques. This paper focuses on how routing data can be used for *reactive detection and identification of* network disruptions as summarized in Figure 2.

Detection with network-wide analysis of routing dynamics. We first collect the set of routing updates from every router in a single domain and perform a multivariate analysis on this set of timeseries data to identify disruptions. We show in Sections 4 and 5 that many network disruptions cause events that exhibit network-wide correlations in BGP routing streams; multivariate analysis helps identify the events that appear simultaneously in many routing streams but do not appear significant from any single routing stream.

Identification with network-wide hybrid analysis. After detecting failures in groups of routing streams, we analyze the nature of these changes by examining the semantics of the routing messages in the context of the model of the network configuration. This process allows us to extract information from the network about the BGP-level connectivity both inside the network and between the local network and its neighbors (*e.g.*, which routers in the network connect to a given neighboring AS).

As we describe in more detail in Section 2, previous efforts to analyze routing dynamics have typically conflated these two tasks by assuming that noise in routing data always implies the existence of a network disruption and applying inference techniques to groups of routing messages to help localize failure causes. In contrast, we address these two problems separately.

2.2 Detection with BGP Routing Data

Both traffic and routing data provide information to network operators about the performance of a running network. Although traffic data often provides more direct information about the performance of individual traffic flows as they traverse the network, routing data can both provide information about systemic network disruptions that affect many traffic flows and offer clues as to *why* a particular disruption is occurring. In other words, routing can assist operators in both detection and identification.

Routing data in ISP networks typically comprises both BGP and Interior Gateway Protocols (IGP) (*e.g.*, OSPF [21], IS-IS [24]) data. Although both types of routing protocols offer information about how network conditions change, IGP routing data typically only contains information about internal topology changes; BGP routing data, on the other hand, typically reflects both internal network changes and changes on the network periphery (which exhibits more instability, as we will describe in more detail in Section 3.1).

Unfortunately, routing data—and, in particular, data from the BGP—is notoriously difficult to use for these purposes because (1) it is noisy (*i.e.*, many routing messages reflect changes in network conditions but not actual *actionable* network events), and (2) the routing messages themselves carry little or no information about the source of a problem. Internet-wide root cause analysis has proven difficult (if not impossible), as we discuss in Section 8.3.

2.3 Single-Network BGP Routing Dynamics

This section provides an overview of single-network routing dynamics. We offer the reader intuition for why routing data should exhibit network-wide correlations upon changes in network conditions such as link, node, or protocol session failures. We consider three types of disruptions that are local to a single network:

1. Link. A link disruption that is internal to the network, as shown in Figure 1(a), can result from the physical failure of a link (or a component on either end of the link), maintenance or re-provisioning, or the disruption of the routing protocols running over that link (*i.e.*, the internal routing protocol or internal BGP session). These failure modes can cause different types of correlated events to occur. For example, Teixeira *et al.* observed that changes to the internal topology due to either link failures or changes in link weights may cause BGP routers to some destinations to change the router that they use to exit the network to one or more destinations (*i.e.*, the egress) [28]. The *number* of routing updates caused by a link disruption will vary depending on the session that is disrupted and the number of destinations (*i.e.*, IP prefixes) being routed over that session. However, link disruptions will typically cause *correlated* events, because they cause many routers in the network to change egress routers.

2. Periphery ("peer"). Disruptions that occur at the edge of the network (*i.e.*, on sessions or links that connect the local network to neighboring networks) can affect how routers inside the network route traffic to external destinations. For example, Figure 1(b) shows an example involving a single link failure that causes multiple routers in the local network to change the egress router that they select en route to some destination. As with link failures, this type of session failure causes correlated routing events across the network, although, again, the absolute size of events may vary.

3. Node. As with link failures or disruptions, node disruptions can cause many other routers in the network to re-route traffic both to internal destinations and to external destinations (*i.e.*, via different egress routers). These disruptions are usually visible across multiple streams of BGP routing data. As we describe in Section 3, unplanned outright node failures are relatively uncommon in the Abilene backbone; we expect that node failures are relatively uncommon in general.

3. Data and Preliminary Statistics

This section describes the datasets we used for our study: (1) The Abilene operational mailing list, `abilene-ops-l`, which documents known failures that have occurred on the network and is used as "ground truth" to study how failures show up in BGP and later for validation; (2) BGP updates from all but one of the routers in the Abilene network, which we use to for detection; and (3) routing configurations from the Abilene network, which we use for identifi-

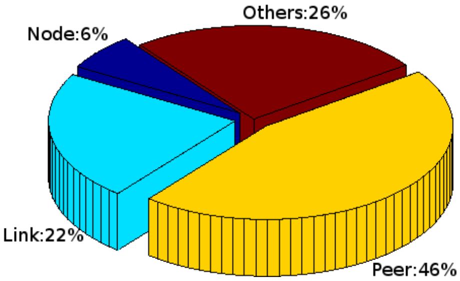


Figure 3: Classes of problems documented on the Abilene network operations mailing list from January 1, 2006 to June 30, 2006 [1].

	instability	unavailability	maintenance	total
node	1	1	22	24
link	0	20	65	85
peer	14	82	77	173
total	14	104	164	282

Table 2: Classes of problems documented on the Abilene network operations mailing list from January 1, 2006 to June 30, 2006.

cation. For the remainder of the paper, we limit our analysis to data collected from the Abilene network because it is the only network where we have access to all three of these data sets.

3.1 Mailing List: Documented Failures

We analyzed documented network disruptions over a six-month period, from January 1, 2006 to June 30, 2006. These documented network disruptions affect three types of network elements—nodes ("node"), internal links ("link"), and peripheral sessions to neighboring networks ("peer")—and can be further classified into three types: instability, unavailability and maintenance. Figure 3 shows the distribution of different events reported via email to the Abilene operational mailing list [1]; the reported events comprise both customer-generated complaints and disruptions detected by the network's automated monitoring system. The mailing list contains multiple emails referring to the same event (*e.g.*, updating the status of a previously reported event) and some other emails regarding network policy. For each actual event, the start time and the end time field in the email are manually entered into the tickets and then reported to the mailing list. We count each event only once and classify all duplicate emails into a class called "others". 97 such events in the six-month period are in this category, which account for 26% of all emails sent to the list.

Table 2 illustrates how many disruptions of each class appeared in our analysis. *Instability* describes problems where network elements go down and come up repeatedly in a short time period. *Unavailability* means that the some network elements are completely offline for some time period. *Maintenance* refers to planned events. In this case, operators send email before the event and perform the action in the reserved time window. Because the time window reserved for maintenance is always longer than the actual event, and because these planned events are likely to be less disruptive, we exclude maintenance problems from our analysis for much of the remainder of the paper (except in Section 6.3, where we attempt to explain various types of "false alarms").

3.2 BGP Updates: Routing Dynamics

Abilene has 11 backbone routers. Each router maintains an internal BGP (iBGP) monitoring session to a collection machine. Because the updates are collected in this fashion, we cannot observe

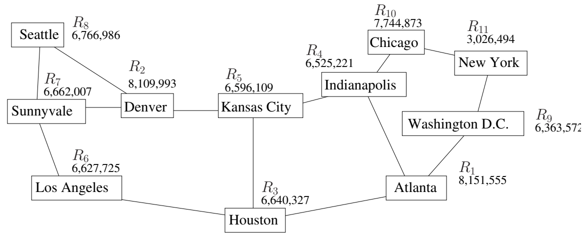


Figure 4: The Abilene backbone network topology, and the total number of BGP updates each router received over the period of our analysis. The figure shows physical nodes and links in the topology but omits iBGP sessions (every router has an iBGP session with every other router) and Abilene's connections to neighboring networks.

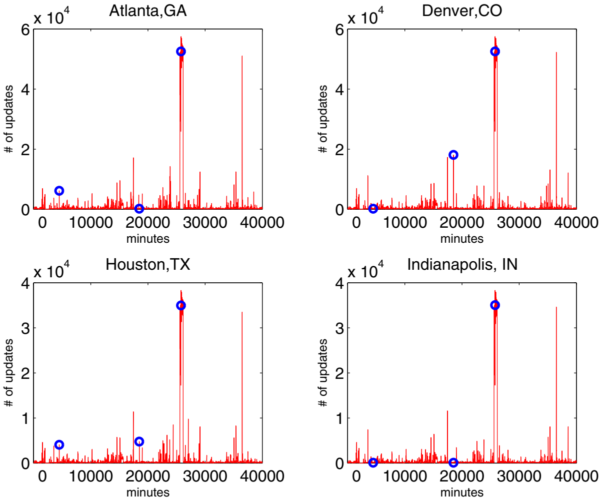


Figure 5: BGP update timeseries data from January 2006 for 4 Abilene routers, with three examples of network disruptions circled.

every BGP update received at each router; rather, we only see the instances when a router *changes* its selected route to a destination. This collection mode is a common way to collect BGP updates in many large ISPs and has been used to analyze BGP routing updates in a single network in other studies [7, 28, 30].

We analyzed the ensemble of BGP update streams from Abilene's routers over six months in 2006, as summarized in Figure 4. We analyzed data from all 11 Abilene backbone routers, with the exception of the router in New York, NY, whose local BGP update monitor failed on February 20, 2006 at 17:39:23 GMT and was not restored for the remainder of our analysis. After collecting BGP update streams for each router shown in Figure 4, we discretize the updates into timebins of 10 minutes; this binsize is a small enough time interval for us to manually inspect the detected events, and it also reduces the likelihood that a BGP pathology resulting from a single network disruption is spread across multiple timebins (previous work observed that most BGP routing pathologies resulting from a single disruption do not last longer than 5 minutes) [14].

Figure 5 shows an example of BGP update timeseries from different routers in the Abilene network during January 2006. The circles on each timeseries mark three examples of documented disruptions. These example disruptions illustrate a fundamental problem with detecting network disruptions using BGP update data: Any single stream of routing updates is extremely noisy; to make matters worse, the number of updates in any time interval does not correlate well with the severity of the event (for reasons we discuss in Section 4.2). Simple threshold-based detection schemes will not detect disruptions accurately. Thus, the task at hand is to *mine* actionable disruptions from this noisy and complex data.

3.3 Configuration Data: Network Model

Abilene makes its routing configurations publicly available. The configurations allow us to obtain: (1) the total number of external BGP sessions that the local network has with neighboring networks; (2) the next-hop IP addresses and neighbor AS numbers of the routers on the other side of those sessions; and (3) the number of eBGP-speaking routers inside the local network and the next-hop IP addresses of those routers. These configurations allow our identification algorithm to incorporate information about the network configuration and topology that helps in identifying the type of network disruptions detected. We describe our identification algorithms in more detail in Section 6.

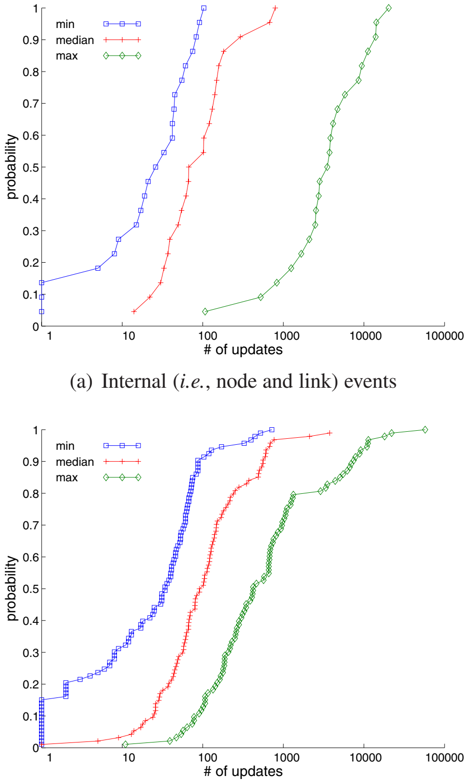
4. Characterizing Network Disruptions

No detection method that relies on BGP updates will detect anomalies that are not visible in BGP. We believe that, before we can even begin to design a method for detecting network disruptions using BGP routing data, we must answer the following question: *To what extent do known, documented network disruptions within a network appear in BGP routing data, as observed from the perspective of that network?* Our approach—which first analyzes "ground truth", documented cases of network disruptions and then searches for evidence of these disruptions in BGP routing data—marks a significant departure from previous work on analysis of BGP data, which works in the opposite direction (*i.e.*, first observing BGP routing data under the assumption that "noise" in routing data implies network disruptions and then searching for the cause of the disruption). This study, on the other hand, analyzes the email list that documents all problems happened in a local network to determine whether these events are visible in BGP and, when they are, how they are reflected in routing messages.

4.1 Most disruptions appear in BGP updates

We first attempted to determine whether the known, documented network events on the Abilene network backbone (summarized in Figure 3 and Table 2) appeared in the BGP routing messages measured on the backbone network. Although this question appears rather basic (and our results are not particularly surprising), it is important to verify, because if network disruptions do *not* appear in BGP, there is no hope of detecting network disruptions with *any* detection mechanism. For example, previous work has noted that only about half of observed end-to-end path disruptions in the wide-area are ever reflected in BGP update messages [5], indicating that wide-area "root cause analysis" of performance problems may miss many important events. Some examples where wide-area network disruptions and routing failures are not reflected in BGP are described in detail in previous work [27].

In contrast, we wanted to gauge the effectiveness of using BGP to detect and identify disruptions *within a single network*. To do so, we analyzed each documented network disruption in the Abilene backbone network on the Abilene operational mailing list [1] and determined whether the documented event was *visible* in BGP. We say that an event is visible in the following cases: (1) for "peer" events, we should see the BGP updates around the time of the reported event included BGP update messages with the neighbor AS of the peer reported to be involved in the disruption; (2) for "node" events, we should see BGP updates on every other router that replace all the old routes through the node that failed; and (3) for "link" events, we generally expect to see a noticeable increase in the number of BGP updates at many routers, since these can cause many routers in the network to change egress routers [28].



(a) Internal (*i.e.*, node and link) events

(b) Peripheral (*i.e.*, peer) events

Figure 6: The number of BGP updates that occurred in any 10-minute interval for events documented in the Abilene operational mailing list [1]. Both internal and external events vary in maximum size over several orders of magnitude, making it difficult to design a general detection scheme based on a single threshold value.

We find that both of the 2 node disruptions are visible in BGP; all but one link event and all but 8 peer disruption events are visible in BGP (Table 3 summarizes these statistics, which we revisit in Section 5 when we discuss the detection of these events). Upon further inspection, we found that these events were not visible in IGP routing data, either. Several events were also reflected in the BGP routing data even though no IGP changes were reflected. This high visibility makes sense, because most internal network disruptions are likely to affect how traffic is routed to external destinations as well. We suspect that the small number of events that are not visible in BGP may even be explainable by factors such as reporting errors to the mailing list.

4.2 Disruption sizes are highly variable

Various factors can affect the number of routing messages that result from a network disruption. When the disruption occurs at the network *periphery* (*e.g.*, when a peering session fails), the size of the disruption is related to the number of routes being advertised on that particular session. In the case of a node or link failure, the number of BGP updates in the event is more indirect: It is related to the number of routes for which other routers were using the failing node or link to reach the destination. In both cases, the size of the disruption will vary depending on the router in the network where the disruption is being observed. For example, in the case of a peering session failure, the local router will see a disruption for all routes on that session. For the same event, a monitor at a different router will see only BGP updates both for which the border router actually changed its route to the destination (causing that router to see updates) and for which that router also changed its route (causing that router's monitor to see updates).

We *quantify* the distribution of the BGP update bursts that result from network disruptions. Figure 6 shows, for each network disruption, the maximum, median, and minimum number of updates (quantized in 10-minute bins) received across the routers for each documented disruption; each sub-figure shows this distribution for internal and external events, respectively.[1] One point on each of the lines (*i.e.*, minimum, median, maximum) on a CDF represents a single network disruption, which may span multiple 10-minute intervals. For example, for internal network events shown in Figure 6(a), the "smallest" disruption (reading the "bottom" three points from Figure 6(a)) triggered zero BGP updates at one or more of the routers, and a burst of 100 updates in some 10-minute interval at one or more routers; the median burst size across routers for that event in any 10-minute interval was 11 updates.

Figure 6 reveals interesting characteristics of how internal network disruptions are reflected in BGP updates. First and foremost, we note that network disruptions can be reflected in BGP update "spikes" of sizes that span several orders of magnitude. Second, for internal events, the maximum number of updates is much larger than the median and minimum number of updates; this makes sense: internal events are likely to cause many updates in a few routers, but destinations for which routers are not using the failed node or link will not be affected. For example, in the case of external events, 80% of network disruptions have a maximum burst size of less than 1,000 updates at any single router; furthermore, the maximum burst size at some router in the Abilene network for any external network event spans about four orders of magnitude (from 10 updates to more than 10,000 updates).

This wide variation in the size of a network disruption underscores the difficulty in detecting network disruptions solely on their sizes. Fortunately, as the next section illustrates, these network disruptions typically exhibit *network-wide correlation—i.e.*, simultaneous disruptions at more than one router—regardless of their size.

4.3 Disruptions have spatial correlation

Network disruptions of all types almost always elicit some routing updates at more than one router across the network. Figure 7 shows examples of each of the three types of network disruptions we study and how they appear in the BGP routing data. In each case, as we show in the previous section, the actual number of routing updates induced varies widely both across events (some events cause thousands of updates, while others cause tens or hundreds), and also across routers for any single event. In all three examples, though, *all routers experience some disruption at nearly the same time*. For the reasons described in Section 4.2, a single network disruption may affect not only the routers that are directly involved but also other routers in the network.

We first study three specific examples of network disruptions and the resulting routing messages seen across the network:

Node disruption. Figure 7(a) shows the BGP updates that result at each router in Abilene when the Abilene backbone router in Houston became unavailable on May 19, 2006 from 11:01 p.m. to 11:08 p.m.. The monitor at the router that experienced the failure, R_3 (Houston), sees an abnormally large number of BGP messages; other routers do not see such large spikes at this time, but all routers witness *some* disruption.

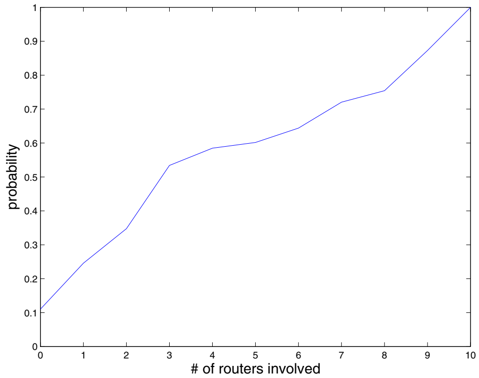


Figure 8: Distribution of number of routers experiencing BGP updates associated with a single network disruption.

Link disruption. Figure 7(b) shows a disruption on the link between routers in Denver and Seattle, which became unavailable from 4:56 p.m. to 5:03 p.m. on April 11, 2006. In this case, three routers—R_2 (Denver), R_6 (Los Angeles) and R_7 (Indianapolis)—see about 600 routing changes, while the rest of the routers in Abilene experience far fewer updates.

Peer disruption. Figure 7(c) shows a network disruption where various peers connecting to the Abilene router in Sunnyvale lost connectivity from 8:28 PM to 8:43PM on February 9, 2006. In this case, three routers—R_2 (Denver), R_6 (Los Angeles) and R_7 (Indianapolis)—see two update spikes of about 400 updates each and other routers see two small spikes of about 100 updates each. We have shown three specific example network disruptions where disruptions give rise to routing updates that are correlated across routers; we find that the correlation of routing updates across the routers in a single network holds in general. Figure 8 shows a CDF of the number of routers at which any particular network disruption was "BGP visible" (as we defined in Section 4.1). When we say a network disruption reported is not visible in BGP, then this is equivalent to cases where the disruption is visible at zero routers. Strikingly, more than 75% of network events are visible in BGP at more than one router in the network. This high occurrence of correlation *across* routers suggests that performing multivariate analysis over collections of routing streams in a single network may be effective for detecting network disruptions; we explore the feasibility of this approach in the next section.

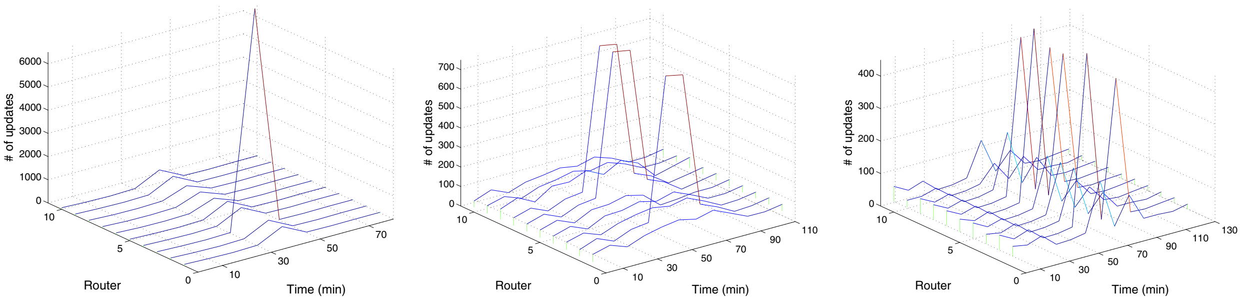
5. Detecting Network Disruptions

The previous section demonstrated that network disruptions typically exhibit network-wide correlation across streams of BGP update messages, even though individual streams of update messages can vary in size depending on the actual network element affected by the disruption and the number of destinations (*i.e.*, IP prefixes) being routed through that network element.

5.1 Network-Wide Analysis

The results from Section 4 indicate that known network disruptions do appear in routing data as correlated update streams. Many multivariate analysis techniques can be used to extract such dependencies from the ensemble of update timeseries. One such technique is the *subspace method*. The subspace method draws on ideas

[1] Because the number of node disruptions is so small (only two events occurred over the six-month period of our analysis), we classify both node and link disruptions internal events.



(a) Node Disruption

(b) Link Disruption

(c) Peer Disruption

Figure 7: Examples of how three types of network disruptions appear across 10 Abilene routers. The index on the y-axis indicates the router's ID from Figure 4; for example, 1 is router R_1 (Atlanta). These examples illustrate that, though the magnitude of updates that may induce a variable number of updates (thus making threshold-based detection difficult), multiple routers in the network will often witness some evidence of the disruption.

from multivariate statistical process control [4], and has been previously used to detect anomalies in timeseries of network-wide traffic counts [16, 17]. We first introduce notation and then briefly review the main ideas of the subspace method from [16], in the context of network-wide feeds of BGP updates from multiple routers.

Let $\mathbf{X}$ denote a $t \times r$ matrix ($t \gg r$), where t is the number of time bins and r is the number of routers. Each element of this matrix, x_{ij} denotes the number of BGP updates in router j at time i. Each column j of this matrix is the timeseries of the number of BGP updates seen at router j.

The subspace method performs a transformation of basis to separate the multivariate timeseries into normal and anomalous temporal patterns. Normal patterns are those that are most common temporal trends in $\mathbf{X}$: together they capture a dominant fraction of variance of $\mathbf{X}$. These common patterns are extracted by decomposing $\mathbf{X}$ via Principal Component Analysis (PCA); previous work has performed similar analysis on network traffic [18]. PCA decomposes the collection of update timeseries into their constituent temporal patterns, such that these trends are ranked according to the amount of variance they capture in the original data. Due to the strong network-wide dependency in the BGP update streams across the routers in the network, we find that the top 2-4 temporal patterns capture the vast majority of the variance (90%) in the update timeseries. The subspace method uses this ordering to designate the temporal patterns that account for a large fraction of the total variance as constituting the *normal subspace*, and all remaining trends as being the *anomalous subspace*.

After the subspace method computes normal and anomalous subspaces, each router's timeseries can be expressed as a linear combination of normal and abnormal components, by projecting each router's timeseries onto each of the two subspaces. Specifically, we can express the number of updates seen by all the routers at a particular point in time ($\mathbf{x}$), as the sum of normal and residual components, *i.e.*, $\mathbf{x} = \hat{\mathbf{x}} + \tilde{\mathbf{x}}$. Here, $\hat{\mathbf{x}}$ is the reconstruction of $\mathbf{x}$ with only the normal temporal patterns, and $\tilde{\mathbf{x}}$ contains the remaining temporal patterns. Anomalies by the subspace method are detected by inspecting the size of residual vector ($\|\tilde{\mathbf{x}}\|^2$) across time for unusually large values. In particular, an anomaly is triggered when $\|\tilde{\mathbf{x}}\|^2 > \delta_\alpha$ where δ_α denotes the Q-statistic at the 1-α confidence level, as given in [10] and used for traffic analysis in [16]. We set α to be 0.001, which puts detection at the 99.9% confidence level.

5.2 Design and Implementation

Our detection system is implemented in three phases: collection and database insertion, post-processing, and analysis. Although our detection system currently performs only offline analysis, we believe that it could be extended to perform online analysis without fundamental modifications to the architecture. Our system periodically collects Abilene BGP update data that is logged by the Abilene BGP monitors, as described in Section 3.2; we then process these files and insert them into an SQL database, which also contains the network representation from the network's routing configurations that we use for identification (described in Section 6). Insertion of one day's worth of Abilene routing data (the granularity at which we were inserting batches of routing messages) takes less than 5 minutes, including building the database indexes for that data. The collection and data processing modules are implemented in about 800 lines of Perl and Ruby.

We have implemented a BGP update post-processor that groups BGP update timeseries data into timebins of arbitrary size outputs matrixes for input to our implementation of the subspace method. The update post-processor is implemented in about 550 lines of Ruby, and our implementation of the subspace method is about 70 lines of Matlab and processes a 200 × 11 BGP update timeseries matrix (*i.e.*, the number of routers, times about 1.5 day's worth of 10-minute timebins) in an average of 22.7 milliseconds on a 2.80GHz processor with 4GB of RAM. (We show in the next section that this amount of routing data is reasonable for detecting network disruptions using the subspace method.)

5.3 Results

In this section, we quantify the effectiveness of using multivariate, network-wide analysis of routing updates to detect network disruptions that might otherwise be missed. In particular, we find the following: (1) the subspace method detects every documented link and node failure on the Abilene backbone network and nearly two-thirds of documented failures on Abilene peering links; (2) the amount of routing data that must be processed to successfully identify network disruptions are reasonable, suggesting that our techniques could ultimately be incorporated into an online detection system; and (3) though specific parameters in the subspace method are tunable, the technique works well for a wide range of settings. Our evaluation should not be read as the last word on tuning a specific algorithm (*i.e.*, PCA) to detect network events; indeed, there are many other angles to explore in terms of network-wide analysis (*i.e.*, different multivariate analysis algorithms, different input timeseries, etc.), which we discuss further in Section 9.

In this section, we quantify how well the subspace method detects the network disruptions that are visible in BGP, and how well it detects events of various magnitudes. Based on our characteriza-

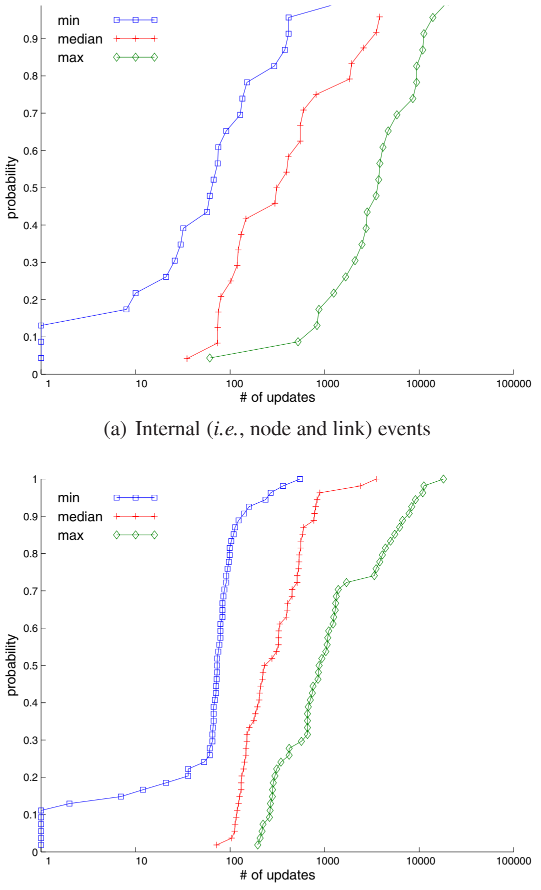
	Instability and Unavailability	Visible in BGP	Detected by PCA	Rate
node	$1 + 1 = 2$	2	2	100%
link	$0 + 20 = 20$	19	19	100%
peer	$14 + 82 = 96$	89	54	60.67%

Table 3: Number and fraction of network disruptions of each type detected by the subspace method.

tions of how network disruptions are reflected in BGP update messages in Section 4, we hypothesized that a multivariate, network-wide detection scheme would be effective at detecting network disruptions, (because these disruptions exhibit correlations across routers) and, further, that such a scheme could even do well at detecting network disruptions that did not generate a large number of updates. We find that, over the duration of the six months of our study, the subspace method detects all documented node and link disruptions on the Abilene backbone and about two-thirds of documented failures on peering sessions. Furthermore, we find that the subspace method detects many network disruptions that do not generate a large volume of updates at any single router; this finding highlights the strength of the subspace method, which can detect events that exhibit correlation across routers, even if they do not generate a large "spike" in any single routing stream.

Detection rate. Table 3 illustrates the number of documented disruptions that are visible in BGP and, among those, the number and percentage of disruptions that the subspace method captures. The subspace method detects every disruption to an internal node or link and about 60% of disruptions to peering links, at a cost of a reasonable rate of "false alarms": our method generates an average of no more than three false alarms per day.[2] Figure 9 shows the maximum, median, and minimum number of updates received in a ten-minute interval by any router in the Abilene backbone network for all internal and external events detected by the subspace method and demonstrates that the subspace method can detect network disruptions even when these disruptions do not induce a large number of routing updates.

We wondered why the subspace method was less successful at detecting disruptions on peering links at the network periphery. We found that the 35 peering sessions that the subspace method failed to detect all had two common characteristics. First, these events only caused a significant number of updates at a single router. Second, even the disruption that occurred was very small, often causing far less 100 routing updates at any single router. It makes sense that multivariate analysis fails to detect events that do not exhibit network-wide correlation. It is also reasonable that peering events, which occur outside the network, may not give rise to correlated events within the network, especially if the sessions do not carry many routes or if they do not cause many routers in the network to update their selection of routes.



(a) Internal (*i.e.*, node and link) events

(b) Periphery (*i.e.*, peer) events

Figure 9: The number of BGP updates that occurred in any 10-minute interval for events documented in the Abilene operational mailing list [1], *for all events detected by the subspace method*. The subspace method is capable of detecting a significant fraction of low volume events, particularly the "BGP-visible" internal events (for which it has a 100% detection rate, even for low-volume events).[4]

	median	minimum
node	43.35	29.17
link	57.87	14.38
peer	96.13	9.25

Table 4: Delay (in minutes) for the time each network disruption was reported to the Abilene mailing list from the time it actually occurred, for the three types of disruptions.

Detection time. Our detection algorithms might also be able reduce detection time. To gain some intuition about just how much our methods could reduce detection time, we study the delay from when the subspace method detected a network disruptions on the Abilene network to the time when it was actually reported to the Abilene operational mailing list. Although much of the Abilene outage reporting to the operational mailing list is automated, we recognize that there is often inherent delay in reporting events to mailing lists. Therefore, our study should be considered as an informal indication that there is room for improvement in reducing detection time. Table 4 shows the results of this experiment; the shortest delay we observed for any type of network disruption other than maintenance was 9 minutes, which indicates that even our simple detection techniques could reduce the time to detection.

Effects of parameter settings. To evaluate how the performance of the subspace method was affected by various parameter settings, we evaluated its detection rate for various window sizes. We selected a window size of 200 10-minute intervals as a default for

[2] We cannot precisely determine the rate of false alarms because the operational mailing list is not guaranteed to capture all disruptions to the network; therefore, an event detected by our techniques that is not reported to the mailing list may simply represent an undocumented failure. In Section 6.3, we aim to better understand the alarms raised by the subspace method that do not correspond to documented disruptions.

[4] Figure 9(a) considers statistics over only the interval where the subspace method detected an event, so it differs slightly from Figure 6(a), which considers statistics over the time interval documented on the mailing list.

window size (bins)	node	link	peer
100	1	17	57
200	2	19	54
300	2	18	45
400	2	17	39

Table 5: Number of each type of disruption detected by the subspace method using different window sizes. In all cases, the size of one timebin is 10 minutes, so 100 timebins represent a time interval of just under 17 hours. The rest of our experiments (*e.g.*, the results from Table 3) use a default window size of 200, but our experiments indicate that the algorithm is relatively insensitive to this parameter.

our experiments, but we also evaluated our detection method for other window sizes. The results in Table 5 show how the detection rate for peer events changes for different window sizes; they also illustrate that the subspace method is effective at detecting network disruptions for various window size settings and that our method is relatively insensitive to the exact value of this parameter.

6. Identifying Local Network Disruptions

In the last section, we demonstrated that multivariate analysis techniques are effective at *detecting* network disruptions, but network operators need to know not only that a significant network disruption occurred but also more information about the likely cause of that disruption. In this section, we present the design, implementation, and evaluation of a simple heuristic that identifies the type of network disruption that occurred. We call our general approach *hybrid analysis* because it uses a combination of static analysis of router configuration files and analysis of the routing updates to identify the type of failure.

Although our approach bears some similarities to the BGP anomaly classification in previous work [30], it has several significant differences. First, this previous work described various disruption scenarios in terms of their effects (*e.g.*, internal path change) but did not propose an algorithm for determining the reason for the changes (*e.g.*, node failure). In contrast, we propose a *prescriptive algorithm* for identifying the type of network disruption (*i.e.*, node, link, peripheral, or external) and implement this algorithm to process the static routing configurations and dynamic BGP routing data. Second, we *validate* our identification algorithm using "ground truth" information about network disruptions from the Abilene backbone network to verify that our identification algorithm is correct. classifier correctly identifies every node and link disruption and 93% (28 of 30) of the detected peer disruptions. Finally, as we describe further in Section 7, our hybrid analysis approach is general: we explain how it could be used not only for identifying the type of disruption that occurred but also to help identify the actual location of the disruption within the network.

6.1 Network-Wide Hybrid Analysis

Our identification algorithm builds a network model from the static router configuration files and uses this model with the next-hop attribute in the routing updates to distinguish different network disruptions. Our current algorithm only differentiates the *type* of network disruption without actually locating the actual network element that failed; this heuristic only requires the IP addresses of the routers within the network and the IP addresses of the opposite ends of the external BGP sessions (*i.e.*, the IP addresses of the routers in neighboring networks with peering BGP sessions to the local network). Section 7 describes our ongoing work to precisely locate disruptions within the local network, a task which requires additional information from the routing configuration files.

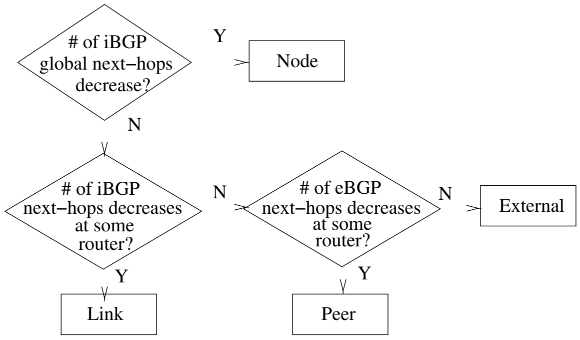


Figure 10: Decision tree for identifying network disruption types.

Figure 10 describes the algorithm we use to identify the type of network disruption that occurs on the Abilene network. The identification algorithm maintains and tracks three features: (1) the total number of next-hop IP addresses selected by all routers in the network ("global internal BGP (iBGP) next-hops"); (2) at each router, the distinct number of next-hop IP addresses selected by that router to other routers within the network ("local iBGP next-hops"); and (3) at each router, the number of distinct next-hop IP addresses selected by that router to other routers outside the network ("local external BGP (eBGP) next-hops"). The first feature allows the algorithm to determine how many routers in the network are currently being selected as the egress router from the network; if this number decreases universally for all routers, the likely explanation is that some node in the network has failed. The second feature tracks the number of other routers within the local network that each router is selecting as an egress router; if this count decreases at some router, but does not decrease for the entire network, our algorithm infers that an internal link has failed. This rule is also fairly intuitive—if one node becomes unreachable from another, it will often stop selecting that node as an egress router. We apply similar reasoning for the third phase of identification, which identifies disruptions at the periphery of the network—which typically affect whether some router selects some router in a neighboring network (and thus affects the number of eBGP next-hops at that router). Our inference algorithm does not identify link and peer disruptions perfectly, but the algorithm is more than 80% accurate for all types of failures, it is simple to implement, and it is computationally efficient. We discuss our validation in Section 6.3.

6.2 Design and Implementation

The identification algorithm is implemented in two phases: a bootstrapping phase, where the algorithm constructs the routing tables for each router in the network and computes initial values for the three features that it tracks; and a run-time tracking phase, where the algorithm maintains the sets of iBGP and eBGP next hops both for each local router and globally for the network. All BGP data is maintained in the SQL database described in Section 5.2; we use this update data to derive a new table, which keeps track of changes to the sets of next-hop IP addresses over time; this derived table will allows the system to issue a query for a specific time (*i.e.*, the time of the detected event) and determine whether the cardinality of any of the three next-hop sets changed around the time of the failure. Additionally, we use the publicly available rcc tool [6] to parse the routing configurations to glean information about which next-hop IP addresses are internal vs. external, which routers in the network have sessions with which next-hop IP addresses, etc. The algorithm for deriving this auxiliary data is implemented in about 100 lines of Perl and can process one day's worth of BGP update data in about 5-10 minutes, depending on the

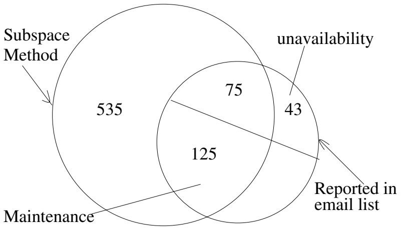


Figure 11: Results for identification over six months of operation. The 535 detected events (about three per day) may either be "false alarms" or disruptions that were undocumented on the Abilene mailing list[1].

volume of routing updates for a single day. The tracking phase is implemented in about 30 lines of Perl.

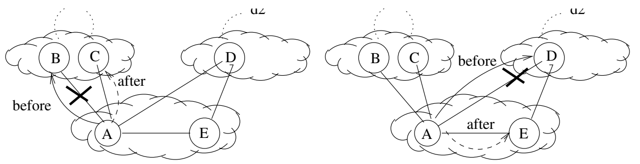
6.3 Validation Results

In this section, we validate the identification algorithm from Section 6.1 (Figure 10). Our goal is two-fold. First and foremost, we seek to evaluate the correctness of our algorithm by comparing its results against the network disruptions for which we have "ground truth" documentation about the type of disruption that occurred (*i.e.*, from the Abilene operational mailing list [1]). Second, we aim to understand as best we can the network events that the subspace method detected but were not network disruptions.

To validate our identification algorithm, we applied the algorithm shown in Figure 10 to every network disruption that was detected by the subspace method. For the 75 disruptions that were documented as instability or unavailability and we detected with multivariate analysis, we checked whether the output of our identification algorithm agreed with the type of disruption that was indicated on the mailing list. Of these disruptions, our algorithm successfully classified both node disruptions, 14 of 19 link disruptions (74%), and 28 of the 30 peer disruptions (93%) where we have BGP update data from all eleven routers.[5]

We examined in closer detail the 2 peer disruptions and 5 link disruptions that were mis-identified as external events. We believe that it is entirely possible that the two misclassifications for peer disruptions are due to reporting mistakes to the mailing list: a close examination of the BGP data shows absolutely no activity for the neighboring networks listed as being involved in the disruption. The reasons for misclassifying the link disruptions appear to be more subtle and include multiple possibilities. In some cases, it appears that the duration of the link failure is extremely short; in these cases, it is possible that the routers did not update their iBGP next-hops to another router before the link was restored. We believe that refinements to our identification algorithm—perhaps by incorporating additional data sources (*e.g.*, internal routing data)—may help us disambiguate these few ambiguities. Another possibility is to relax the rules in the existing algorithm: rather than requiring the number of iBGP next-hops to drop to zero to declare a link failure, identifying a link failure based on a sharp drop in the number of routers selecting a particular iBGP next-hop may also help correctly identify these cases.

We also perform identification on *all* of the events detected by the subspace method to better understand some of the events that were detected but not documented on the mailing list. Figure 11 summarizes the events detected by the subspace method and their relationship to the set of known, documented disruptions. The subspace method detected a total of 735 events: 75 of which are known



(a) External next hops only.

(b) Internal and external next hops.

Figure 12: An example where a combination of static and dynamic analysis can help localize disruptions. Knowledge about internal and external next-hops, and observations of how they change in an update burst can differentiate different cases.

instability and unavailability events. An additional 125 events occur within documented maintenance intervals, suggesting that these detected events very likely correspond to maintenance-related disruptions. As previously discussed in Section 5.3 (Table 3), the subspace method fails to detect 43 disruptions related to instability and unavailability, most of which are disruptions to peering sessions, as opposed to internal node or link disruptions.

The subspace method also detects an additional 535 events; although these events are not documented failures, *we cannot necessarily consider all of them to be false alarms*. Because the Abilene mailing list only documents disruptions that the current detection systems are capable of detecting, it is possible that some of the events that the subspace method detects are actually previously undetected network disruptions. We also manually investigated a random subset of 120 of these events, all of which showed some notable BGP activity: found that about 60% have low-volume update bursts that appear at more than one router, about 35-40% are high-volume correlated spikes, an the remainder are big spikes on one router. Without "ground truth" data for these events, we cannot identify the causes of this activity with certainty. Even in the unlikely worst-case scenario, where all 535 events are all false alarms, the average false alarm rate is still only about 3 per day, which is well within the realm of manageability.

7. Towards Isolating Local Disruptions

Our identification heuristic in Section 6 accurately identifies the types (*i.e.*, node vs. link) and general locations (*i.e.*, internal vs. external) of network disruptions, but it does not help a network operator identify a specific failure scenario (*e.g.*, *which* link within the network or at the periphery experienced a disruption). Previous work has made significant advances in identifying which link or node has failed on a global scale [3, 8, 31], and we do not attempt to tackle this task in our work. On the other hand, our preliminary results indicate that isolating the cause of failures *within a single network* may prove to be tractable, given that a network operator has very detailed information about the local network.

We believe that extensions to the approach in Section 6, which jointly analyzes the semantics of the routing updates and the static routing configurations to identify network disruptions, can be extended to help operators identify the location of a disruption, as well as its type. For example, Figure 12 shows two failures at the network periphery that a network operator could pinpoint with knowledge from the routing configuration about the next-hops and neighboring networks that connect to each router. For example, in Figure 12(a), the burst of BGP routing updates would contain only next-hop IP addresses of routers *outside* the network, as router A changed its next-hop route selection from router B to router C. On the other hand, the failure scenario in Figure 12(b) would cause the monitor at router A to see BGP routing messages with next-hops *inside* the local network, as router A changed its route selection from a route with the next-hop outside the local network (router C)

[5] Recall from Section 3.2 that we are missing BGP update data from the Abilene router in New York after February 20, 2006 (about four months of data from that router). Although we detected a total of 54 peer disruptions, 24 of these disruptions occurred that concerned the New York router, so we are missing the data that would help us make those identifications.

to one inside the same network (router E). With knowledge of both the network configuration and the nature of the next-hop changes in the BGP update bursts, an identification algorithm could help localize this network disruption.

8. Related Work

In this section, we survey related work on analysis of routing dynamics in three areas: (1) routing dynamics in a single network, (2) Internet-wide analysis of routing dynamics for "root cause analysis", and (3) the effects of routing dynamics on end-to-end path performance. We emphasize the distinction between our work, which studies network-wide correlations of routing dynamics in a single network to diagnose disruptions, and previous related work, which has largely focused on analysis of single routing streams.

8.1 Single-Network Routing Dynamics

Wu *et al.* proposed a method for analyzing routing dynamics from multiple routing streams within a single network to provide alerts for disruptions [30]. As in other previous work [3, 8, 15], this detection algorithm clusters BGP update messages along three dimensions according to time, prefixes, and views but does not incorporate network-wide dependencies in routing data to improve detection of network disruptions.

Previous techniques for analyzing routing dynamics in a single network can detect network events that affect a large number of Internet destinations or a large amount of traffic, but they have several shortcomings. First, most existing techniques (including that of Wu *et al.*) are threshold-based: they involve setting "magic numbers" for many aspects of network events, including the typical time length of an update burst and the magnitude of the update burst. Second, previous work has shown that clustering updates according to prefixes can occasionally lead to incorrect conclusions about the cause of a network disruption [27]. Rather than grouping routing updates *a priori* based on assumptions about how a specific routing protocol or network configuration behaves, our detection methods are based on analysis techniques that can extract network-wide dependencies but avoid imposing any specific set of assumptions.

8.2 Learning-Based Anomaly Detection

Learning-based approaches have been applied to routing anomaly detection in limited contexts. Previous work has noted the difficulty in setting magic numbers in detection algorithms that rely purely on analyzing the volume of BGP routing updates and has proposed building a model of normal behavior using unsupervised learning. One such method relies on wavelet-based clustering of update volumes to detect abnormal routing behavior [32]; similar wavelet-based decomposition techniques have been used for detecting anomalies in network traffic [2].

Our work is inspired by existing techniques that use multivariate analysis to extract structure from network traffic data [18] and for using these techniques to build models of normal traffic behavior and detect deviations that represent anomalies in data traffic [16, 17, 18]. At first brush, one might view this paper as a relatively straightforward application of these techniques applied to routing data, rather than traffic data, but, as our results in later sections demonstrate, diagnosing routing disruptions requires incorporating a considerable amount of domain-specific knowledge to complement statistical detection.

8.3 Internet-Wide Root Cause Analysis

Xu *et al.* have analyzed BGP routing data using Principal Component Analysis to determine sets of ASes that are affected by the same network event [31]. Their work pioneered the approach of using multivariate analysis techniques on routing data, based on the observation that, because the Internet has structure at the AS-level, a single network disruption can give rise to groups of seemingly unrelated routing updates in different ASes. We apply the same insight to the analysis of routing dynamics *within a single network*. (Others have made similar observations about failures inducing correlated network data streams both at layer 2 [11] and at the IP layer [20].) Xu *et al.* extract correlations from a *single update stream* in an attempt to find structure on an AS-level granularity on the global Internet; in contrast, we analyze *multiple routing streams* from a single network in an attempt to detect and isolate network disruptions within that network. The goals of Xu *et al.* center around "root cause" analysis of Internet-wide dynamics and extracting AS-level structure; in contrast, we focus on diagnosis of network disruptions within a single network.

Our work differs from previous work on "BGP root cause analysis" [3, 8], which analyzes Internet-wide routing dynamics from public vantage points (*e.g.*, RouteViews [23]) to detect Internet-wide events (many of which are artificially injected with "BGP beacons" [19]) and attempts to identify the network that is responsible for causing the update. In contrast, our analysis techniques help an operator of a *single network* detect when network events happen inside that network and identify the cause of the disruption.

8.4 Network Dynamics and Path Performance

Various projects have studied routing dynamics and attempted to characterize and classify them. Previous work has studied BGP routing instabilities and attempted to classify failures based on the observed properties of BGP update messages [9, 12, 13]. Govindan *et al.* found that BGP routing instability was exacerbated by the growth of the Internet [9], and Labovitz discovered that BGP converges very slowly upon a network failure, and that convergence was slowed by path exploration [12]. Both of these projects analyzed single routing streams in isolation and equated BGP instability with network failures but did not study how BGP routing instability correlated with documented network disruptions. Several existing commercial products monitor routing, traffic, or SNMP data for faults, but they typically produce noisy reports about events from the perspective of a single network device [22, 25]. Network-wide analysis of routing data may help operators both identify the severity of these alarms and correlate them, reducing the overall volume of alarms that operators need to process. More recently, various studies have studied how end-to-end path performance correlates with BGP routing instability [5, 29], but, as in previous work, these studies analyze single streams of routing messages that are propagated across the Internet; in contrast, we study correlation across multiple streams of BGP routing messages as observed from different vantage points within the same network.

9. Conclusion

This paper has demonstrated the promise both of using network-wide analysis to improve detection of network disruptions and of using static configuration analysis to help identify the cause of a network disruption. Our analysis techniques represent a new approach to analyzing routing data. Rather than attempting to diagnose disruptions based on temporal fluctuations in a single routing stream, we recognize that (1) the structure and configuration of the network introduces dependencies that give rise to correlated events in groups of routing streams when a network disruption occurs; and (2) this network structure and configuration can be mined to construct a model to better identify the nature of a network disruption.

We have studied the characteristics of how network disruptions induce BGP update messages across the routers in a network backbone over a six-month period and found that, while network disruptions induce routing updates that can vary in volume by several orders of magnitude, nearly 80% of network disruptions exhibit some level of correlation across multiple routers in the network. Based on this observation, we applied the subspace method, a multivariate analysis technique, on BGP update streams across the Abilene backbone. We find that it successfully detects all node and link failures and two-thirds of failures on the network periphery, while keeping the overall alarm rate to an average of roughly three alarms per day. The subspace method performs well for reasonably sized data sets and minimal parameter tuning and, further, that it can process the network-wide routing data in a relatively short amount of time, which suggests that similar multivariate techniques could be incorporated into an online detection and identification system.

We hope that, rather than being the last word on using network-wide analysis to diagnose network disruptions, this paper opens a new direction for exploring a variety of techniques that exploit knowledge of network structure and configuration to jointly analyze sets of network data streams that are inherently dependent. Indeed, many extensions to our work are possible; for example, while this paper has explored the limits of using BGP update *volumes* to detect network disruptions, other attributes in the routing update messages (*e.g.*, the AS path length, next-hop IP address, etc.) may carry semantics that might improve detection in addition to identification. We also recognize that BGP routing update data is not the only possible input to a anomaly detecting system and much work remains to determine how to mine other network datasets and incorporate them into a system for diagnosing network disruptions. As we continue developing techniques to diagnose network disruptions, we hope to gain a better understanding both for which information that best enables diagnosis and for the limits that the information available from current protocols and architectures fundamentally impose on our ability to diagnose these disruptions.

Acknowledgments

We are indebted to David Andersen and the Datapository project for continuing support, and to Emulab for providing hardware resources. We are grateful to Hari Balakrishnan, Haakon Larson, Morley Mao, Anirudh Ramachandran, Jennifer Rexford, and Jia Wang for very helpful feedback and suggestions. This work was supported by NSF CAREER Award CNS-0643974 and by NSF grants CNS-0626979, CNS-0626950, and CNS-0519745.

REFERENCES

[1] Abilene operational mailing list. `https://listserv.indiana.edu/archives/abilene-ops-l.html`.

[2] P. Barford, J. Kline, D. Plonka, and A. Ron. A signal analysis of network traffic anomalies. In *Proc. ACM SIGCOMM Internet Measurement Workshop*, Marseille, France, Nov. 2002.

[3] M. Caesar, L. Subramanian, and R. Katz. Towards localizing root causes of BGP dynamics. Technical Report UCB/CSD-04-1302, U.C. Berkeley, Nov. 2003.

[4] R. Dunia and S. J. Qin. Multi-dimensional Fault Diagnosis Using a Subspace Approach. In *American Control Conference*, 1997.

[5] N. Feamster, D. Andersen, H. Balakrishnan, and M. F. Kaashoek. Measuring the effects of Internet path faults on reactive routing. In *Proc. ACM SIGMETRICS*, San Diego, CA, June 2003.

[6] N. Feamster and H. Balakrishnan. Detecting BGP Configuration Faults with Static Analysis. In *Proc. 2nd Symposium on Networked Systems Design and Implementation*, Boston, MA, May 2005.

[7] N. Feamster, Z. M. Mao, and J. Rexford. BorderGuard: Detecting cold potatoes from peers. In *Proc. Internet Measurement Conference*, Taormina, Italy, Oct. 2004.

[8] A. Feldmann, O. Maennel, Z. M. Mao, A. Berger, and B. Maggs. Locating Internet routing instabilities. In *Proc. ACM SIGCOMM*, pages 205–218, Portland, OR, Aug. 2004.

[9] R. Govindan and A. Reddy. An analysis of inter-domain topology and route stability. In *Proc. IEEE INFOCOM*, Kobe, Japan, Apr. 1997.

[10] J. E. Jackson and G. Mudholkar. Control Procedures for Residuals Associated with Principal Component Analysis. *Technometrics*, pages 341–349, 1979.

[11] S. Kandula, D. Katabi, and J.-P. Vasseur. Shrink: a tool for failure diagnosis in ip networks. In *Proc. ACM SIGCOMM Workshop on Mining Network Data (MineNet)*, pages 173–178, Philadelphia, PA, Aug. 2005.

[12] C. Labovitz, A. Ahuja, A. Bose, and F. Jahanian. Delayed Internet Routing Convergence. *IEEE/ACM Transactions on Networking*, 9(3):293–306, June 2001.

[13] C. Labovitz, A. Ahuja, and F. Jahanian. Experimental study of Internet stability and wide-area network failures. In *Proc. FTCS*, Madison, WI, June 1999.

[14] C. Labovitz, G. R. Malan, and F. Jahanian. Origins of Internet routing instability. In *Proc. IEEE INFOCOM*, pages 218–226, New York, NY, Mar. 1999.

[15] M. Lad, D. Massey, and L. Zhang. Visualizing Internet Routing Changes. *Transactions on Information Visualization*, 12(6):1450–1460, Nov. 2006.

[16] A. Lakhina, M. Crovella, and C. Diot. Diagnosing network-wide traffic anomalies. In *Proc. ACM SIGCOMM*, Philadelphia, PA, Aug. 2005.

[17] A. Lakhina, M. Crovella, and C. Diot. Mining anomalies using traffic feature distributions. In *Proc. ACM SIGCOMM*, pages 217–228, Philadelphia, PA, Aug. 2005.

[18] A. Lakhina, K. Papagiannaki, M. Crovella, C. Diot, E. D. Kolaczyk, and N. Taft. Structural analysis of network traffic flows. In *Proc. ACM SIGMETRICS*, pages 61–72, New York, NY, June 2004.

[19] Z. M. Mao, T. Griffin, and R. Bush. BGP Beacons. In *Proc. ACM SIGCOMM Internet Measurement Conference*, pages 1–14, Miami, FL, Oct. 2003.

[20] A. Markopoulou, G. Iannaccone, S. Bhattacharyya, C.-N. C. and C. Diot. Characterization of Failures in an IP Backbone. In *Proc. IEEE INFOCOM*, Hong Kong, Mar. 2004.

[21] J. Moy. *OSPF Version 2*, Mar. 1994. RFC 1583.

[22] IBM Tivoli Netcool. `http://www-306.ibm.com/software/tivoli/welcome/micromuse/`, 2007.

[23] U. of Oregon. RouteViews. `http://www.routeviews.org/`.

[24] D. Oran. *OSI IS-IS intra-domain routing protocol*. Internet Engineering Task Force, Feb. 1990. RFC 1142.

[25] Packet Design Route Explorer. `http://www.packetdesign.com/products/rex.htm`, 2007.

[26] SSFNet. `http://www.ssfnet.org/`, 2003.

[27] R. Teixeira and J. Rexford. A measurement framework for pin-pointing routing changes. In *ACM SIGCOMM Workshop on Network Troubleshooting*, pages 313–318, Sept. 2004.

[28] R. Teixeira, A. Shaikh, T. Griffin, and J. Rexford. Dynamics of Hot-Potato Routing in IP Networks. In *Proc. ACM SIGMETRICS*, pages 307–319, New York, NY, June 2004.

[29] 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*, pages 375–386, Pisa, Italy, Aug. 2006.

[30] J. Wu, Z. Mao, J. Rexford, and J. Wang. Finding a Needle in a Haystack: Pinpointing Significant BGP Routing Changes in an IP Network. In *Proc. 2nd Usenix NSDI*, Boston, MA, May 2005.

[31] K. Xu, J. Chandrashekar, and Z.-L. Zhang. A First Step to Understand Inter Domain Routing Dynamics. In *Proc. ACM SIGCOMM Workshop on Mining Network Data (MineNet)*, Philadelphia, PA, Aug. 2005.

[32] J. Zhang, J. Feigenbaum, and J. Rexford. Learning-Based Anomaly Detection of BGP Updates. Technical Report YALEU/DCS/TR-1318, Yale University, Apr. 2005.