Measuring the Effects of Internet Path Faults on Reactive Routing

Nick Feamster, David G. Andersen, Hari Balakrishnan, and M. Frans Kaashoek
MIT Laboratory for Computer Science
200 Technology Square, Cambridge, MA 02139
{feamster,dga,hari,kaashoek}@lcs.mit.edu

Abstract

Empirical evidence suggests that reactive routing systems improve resilience to Internet path failures. They detect and route around faulty paths based on measurements of path performance. This paper seeks to understand *why* and under *what circumstances* these techniques are effective.

To do so, this paper correlates end-to-end active probing experiments, loss-triggered traceroutes of Internet paths, and BGP routing messages. These correlations shed light on three questions about Internet path failures: (1) Where do failures appear? (2) How long do they last? (3) How do they correlate with BGP routing instability?

Data collected over 13 months from an Internet testbed of 31 topologically diverse hosts suggests that most path failures last less than fifteen minutes. Failures that appear in the network core correlate better with BGP instability than failures that appear close to end hosts. On average, most failures precede BGP messages by about four minutes, but there is often increased BGP traffic both before and after failures. Our findings suggest that reactive routing is most effective between hosts that have multiple connections to the Internet. The data set also suggests that passive observations of BGP routing messages could be used to predict about 20% of impending failures, allowing re-routing systems to react more quickly to failures.

Categories and Subject Descriptors

C.2.6 [**Computer-Communication Networks**]: Internetworking; C.4 [**Performance of Systems**]: Measurement Techniques

General Terms

Measurement, Performance, Reliability, Experimentation

1. Introduction

The prevalence of faults in the IP substrate results in frequent performance degradations on Internet paths. These faults occur for a variety of reasons, including physical link disconnection [7], software errors [6], and router misconfiguration [13]. Faults cause path failures (outages), increase the volume of routing traffic, and trigger route oscillations and path fluttering. Their effects are visible to end hosts as broken connections, excessive packet loss, and rapidly varying path quality.

A number of recent proposals to improve the availability of wide-area Internet connectivity use *reactive routing*. In this approach, reactive routing systems measurements of network-layer path characteristics such as reachability, loss, and latency, to choose better paths. Most reactive routing systems use a combination of active probes and passive traffic monitoring to decide which paths are better; the differences are in how they take advantage of alternate paths. Resilient Overlay Networks (RON) [2] and Akamai's SureRoute [15] re-route data packets over an overlay network. These overlays (Figure 1) treat the underlying Internet path between two nodes as a single link, forming a higher-layer path through these nodes. Other systems use routing changes to select between paths at the IP layer [17, 20, 21]. Empirical evidence suggests that such schemes often work well at masking path failures, outages, and periods of extreme congestion [2].

This paper addresses *why* and under *what circumstances* reactive routing is able to overcome path failures by asking three specific questions:

Where do failures appear? To mask path failures, destinations on fault-prone paths must be reachable by alternate paths that fail independently. To determine how well reactive routing can mask failures, we must understand where failures occur in the Internet.

How long do failures last? A routing system that takes longer to react to a failure than the duration of the failure will not meet its goals. To understand how reactive a system must be, we must understand how long path failures last.

How do failures correlate with routing protocol messages? In situations where BGP instability correlates with poor path performance or path failures, BGP messages can serve as an indicator of poor path performance. In these cases, reactive systems could detect failures with fewer active probes than would otherwise be necessary. If BGP instability precedes path failures, reactive routing might even proactively route around some failures before they occur. Using BGP information might therefore reduce the reaction time of reactive routing systems.

To answer these questions, we analyze one year of data collected on a geographically and topologically diverse testbed of 31 hosts. The paths between these hosts traverse more than 50% of the well-connected autonomous systems (AS's) on the Inter-

SIGMETRICS'03, June 10–14, 2003, San Diego, California, USA.
Copyright 2003 ACM 1-58113-664-1/03/0006 ...$5.00.

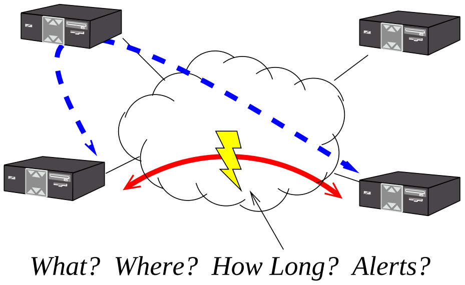


Figure 1: Routing around Internet path failures with overlays. The success of overlays and other reactive routing schemes depends on where failures are occurring and how quickly an alternate path is discovered relative to the duration of the failures. Predicting path failures could allow such systems to pre-emptively route around them.

net. The data includes *correlated probes*, where active probes between hosts discover one-way path failures lasting longer than 2 minutes and trigger traceroutes along these paths when a failure is discovered. We then correlate these observed failures with BGP routing information collected at eight monitoring hosts at the same time.

Our method doesn't pin down where a failure occurs; rather, it captures the location where a failure *appears*. The IP routing substrate reacts to faults it detects by sending routing updates that alter the flow of traffic. Because of this response, the location where a traceroute observes a failure may not be the place where the actual failure occurred and may change with time. Because reactive routing systems must be able to route around where IP routing failures appear rather than where the faults may have originally occurred, this analysis is appropriate for reactive routing systems.

To discover where failures appear, we present techniques for assigning failures to a particular router, and assigning that router to an AS. Using these techniques, we examine where and how long failures appear. Finally, we investigate correlations between path failures and routing instability, as observed from co-located BGP route monitors. Table 1 summarizes the major results in this paper.

In Section 2 we discuss our data collection methods. Section 3 discusses two algorithms that are central to the data analysis: alias resolution and AS assignment, and failure detection and assignment. Section 4 discusses failure location and duration and their effects on reactive routing techniques. Section 5 presents observations of temporal correlations between path failures and BGP messages and suggests how BGP messages could be used to detect and predict path failures. Section 6 surveys relevant related work, and Section 7 concludes.

2. Data collection

We performed measurements between February 2002 and March 2003 on 31 NTP-synchronized nodes in the RON testbed [19], listed in Table 2; during this period, we collected about 60 GB of active probes, BGP messages, and failure-triggered traceroutes. The topology generated by the pairwise paths between these nodes, as measured by traceroute, covers 71 AS's. The testbed topology contains paths that traverse most of the "large" AS's in the Internet. To rank the AS's by size, we counted the degree for each of the 15,040 nodes in the AS graph from the Routeviews table dump of March 13, 2003 at Midnight PST (this technique is a commonly accepted way for approximating the size of an AS [10]). On this date, the paths in our testbed topology traversed 9 of the 11 of AS's that have an AS degree larger than 500 and nearly one-half of the 54 AS's that have a degree larger than 100.

While a few paths are much more failure-prone than others, failures appear spread out over many different links, not just a few "bad" links.	Fig. 5 and 6
Failures *appear* more often inside AS's than on links between them.	Table 7
90% of failures last less than 15 minutes, and 70% of failures last less than 5 minutes.	Fig. 7
BGP messages coincide with only half of the failures that reactive routing could potentially avoid, suggesting that these were failures that not even a "perfect" BGP could avoid.	Table 8
Reactive routing is potentially more effective at correcting failures for hosts with multiple Internet connections.	Sec. 4.3
BGP traffic is a good indicator that a failure has recently occurred or is about to occur. When BGP messages and failures coincide, BGP messages most often follow failures by 4 minutes.	Fig. 13 and 11

Table 1: Summary of major results.

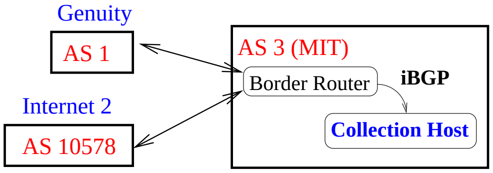


Figure 2: At each collection host, we initiate active probes and collect BGP messages from the network's border router. The figure shows the configuration for MIT, which obtains upstream connectivity from Genuity (AS 1) and the Northeast Exchange (via AS 10578).

We collect data (1) to measure the end-to-end connectivity between hosts using active probes, (2) to determine the location of observed failures using traceroutes to locations found unreachable by the active probes, and (3) to correlate BGP routing changes with failures observed by active probes.

2.1 Active probing

An active probe consists of a `request` packet from the initiator to the target and, if the request gets through, a `reply` packet from target to initiator. Each probe has a 32-bit sequence number, which the hosts log along with the time at which packets were both sent and received. This approach allows us to compute the one-way reachability between the hosts. A central monitoring machine periodically collects and aggregates these logs as described in Section 3.1. Our post-processing finds all probes received within 60 minutes of when they were sent; this margin accounts for clock skew of up to one hour if time synchronization fails.

Name	Location	Probing Start
Aros	Salt Lake City, UT	Feb 13, 2002
AT&T	Florham Park, NJ	Mar 3, 2003
CA-DSL	Foster City, CA	Feb 13, 2002
CCI	Salt Lake City, UT	Mar 6, 2002
Coloco	Laurel, MD	Mar 3, 2003
* CMU	Pittsburgh, PA	Mar 6, 2002
* Cornell	Ithaca, NY	Mar 6, 2002
Cybermesa	Santa Fe, NM	Mar 3, 2003
GBLX-AMS	Amsterdam, The Netherlands	Mar 3, 2003
GBLX-ANA	Anaheim, CA	Mar 7, 2003
GBLX-CHI	Chicago, Illinois	Mar 6, 2003
GBLX-JFK	New York City, NY	Mar 3, 2003
GBLX-LON	London, England	Mar 3, 2003
* Greece	Athens, Greece	May 31, 2002
Intel	Palo Alto, CA	Jul 25, 2002
Korea	KAIST, Korea	Feb 13, 2002
Lulea	Lulea, Sweden	Feb 13, 2002
MA-Cable	Cambridge, MA	Feb 13, 2002
Mazu	Boston, MA	Feb 13, 2002
* MIT	Cambridge, MA	Feb 13, 2002
* **MIT-BGP**	Cambridge, MA	Mar 3, 2003
NC-Cable	Durham, NC	Mar 6, 2002
Nortel	Toronto, Canada	Aug 19, 2002
* NYU	New York, NY	Feb 13, 2002
PSG	Bainbridge Island, WA	Aug 19, 2002
PDI	Palo Alto, CA	Feb 13, 2002
Sightpath	Palo Alto, CA	Feb 13, 2002
* UCSD	San Diego, CA	Mar 3, 2003
* Utah	Salt Lake City, UT	Feb 13, 2002
Vineyard	Cambridge, MA	Aug 19, 2002
VU-NL	Amsterdam, Netherlands	Mar 6, 2002

Table 2: The hosts between which we measured network connectivity. Asterisks indicate U.S. universities on the Internet2 backbone. Hosts at which we also collect BGP data are shown in boldface.

We gathered data on end-to-end connectivity between hosts over a 393-day period between February 13, 2002 and March 12, 2003. The data includes the results from over 390 million probes. To probe, each host independently initiates a probe to another randomly selected host, and then sleeps for a random period of time between 1 and 2 seconds. The mean time between probes on a particular path is 30 seconds, and with 95% probability, each path is probed at least once every 80 seconds. We define failures as three or more consecutive lost probes, which limits the time resolution of our failure detection to a few minutes, rather than seconds.

2.2 Loss-triggered traceroutes

When the active prober declares that the path to a location has failed, the measurement software initiates one traceroute to that location from the location that observed the failure. By keeping track of how far to the destination a traceroute gets, we can obtain an estimate of where the failure manifests itself in the IP topology. We consider the point of failure to be the last reachable IP address. We limit the traceroute to 30 hops.

The failure of a traceroute to reach its destination could indicate either loss of the traceroute probes on the forward path *or* loss of the ICMP time exceeded replies on the reverse path. We use the one-way reachability from the active probes to ensure that the traceroute measurement corresponds to a failure on the forward path.

Each measurement host periodically pushes its probe and traceroute logs to a centralized database server, where the results are joined to match traceroutes to specific path failure events. We have collected over 18,000 loss-triggered traceroutes between June 26, 2002 and March 12, 2003 that coincide with failures.

Host	*BGP Peers*	*Start*	*Updates*
MIT (AS 3)	Genuity, Internet2	Jun 28, 2001	114,424,288
PSG (AS 3130)	Genuity, Verio	May 8, 2002	24,583,211
Vineyard (AS 10781)	Qwest, Savvis	Aug 12, 2002	63,925,991
Nortel (AS 14177)	AT&T Canada	Aug 19 2002	39,964,997
Aros (AS 6521)	UUNet, Electric Lightwave	Sep 2, 2002	47,801,441
GBLX-JFK (AS 3549)	Many ISPs	Jan 27, 2003	11,739,482
GBLX-LON (AS 3549)	Many ISPs	Feb 28, 2003	3,155,999
GBLX-AMS (AS 3549)	Many ISPs	Feb 28, 2003	3,051,377

Table 3: Information about BGP data collected from networks where eight of our testbed hosts are located.

2.3 BGP data collection

The eight testbed hosts shown in boldface in Table 3 collected BGP messages. The hosts ran Zebra 0.92a, an open source software router [24], configured to log all BGP updates. Table 3 also shows the number of BGP updates collected at each site as of March 13, 2003.

Figure 2 shows where the MIT collection host sits in relation to the border router of the hosting network and the rest of the Internet; other monitors sit in similar positions relative to their border routers. MIT's border router has two upstream feeds: a commercial feed via Genuity (AS 1), and a feed to Internet2 via the Northeast Exchange (AS 10578). The monitor receives BGP updates from the border router. Because of the configuration, the monitors will not see all BGP messages heard by the border router; they see only BGP messages that cause a change in the border router's choice of *best* route to a prefix.

Despite not observing all BGP updates, the monitors do observe most BGP messages relevant to routing stability. For example, the MIT border router always prefers routes through Internet2 if they exist. Withdrawal of a prefix on Internet2 would be visible as an implicit withdrawal (i.e., a re-advertisement of that prefix through AS 1). The collection host will not see updates from AS 1 that already have better routes through Internet2, since the best route will not change in these situations.

2.4 Measurement caveats

Several previous studies use traceroute data alone to detect and locate path failures [5, 18]. As we noted earlier, traceroutes alone cannot unambiguously identify one-way outages. Recent work has also shown that traceroutes may not always reflect the path that packets actually take, nor will they necessarily reflect the AS path or where failures occur [1]. Our study combines active probing experiments with traceroutes to address some of these ambiguities.

Traceroute can be filtered by firewalls that block ICMP messages, wrongly giving the impression that the path is faulty. We do not believe this poses a problem with our data because none of our hosts is firewalled. Some routers rate-limit ICMP messages, or handle them on a different processing path than normal packet forwarding. This approach may cause small differences in reachability between traceroutes and normal traffic.

Traceroute identifies the nth hop on a path by setting the TTL of a packet to n, sending the packet towards the destination, and listening for an ICMP time exceeded message from the router at which the TTL expires. Routers *should* set the source ad-

Table	*Description*
	ACTIVE PROBES
probes	Source, destination, request/reply times of UDP probes
failures	Source, destination, start and end times for failures (Derived from *probes*)
downtimes	Host, start/end times when that host was down. (Derived from *probes*)
	TRACEROUTES
topology	Time, source, destination, link (one row per traceroute hop) (daily traceroutes)
topology links	link, source IP, destination IP, estimated depth
failure traceroutes	time, source, destination, link (one row per traceroute hop, triggered traceroutes)
topology routers	Maps IP address (interface) to canonical router (Derived from *topology links*)
router info	Router's estimated border and AS number assignment (Derived from *topology routers*)
	BGP
updates	Time, prefix and mask, update type, attributes
resets	Time at which session resets, reboots, etc. occurred
downtimes	start/end times and name of failed BGP session (Derived from *updates*)
	OTHER
failure details	3-way join of failure, traceroute, BGP messages

Table 4: Database Tables. Tables of measured data are listed in italics, and derived tables are listed in boldface.

dress of the ICMP message to the IP address of the interface on which the triggering packet arrived. Some routers instead set the source address to the IP address of the outgoing interface on the reverse path back to the source [1]. Because we are interested in finding autonomous system boundaries, this inaccuracy is problematic. We therefore use alias resolution techniques developed for the Rocketfuel topology project to accurately assign interfaces to routers [22]. To assign a router to an AS and determine if it is on an AS boundary, we use the method described in Section 3.3. This assignment uses information from both the traceroute in question and the topology snapshots.

3. Analysis approach

We do much of our processing from a central database that contains the measurements from active probes, traceroutes, network topology, and routing data. After describing the database, we discuss an heuristic to approximate the network depth of a link in the topology. We then discuss how to disambiguate traceroute interfaces, assign these interfaces to routers, assign these routers to AS's, and determine whether a router sits on the a network boundary.

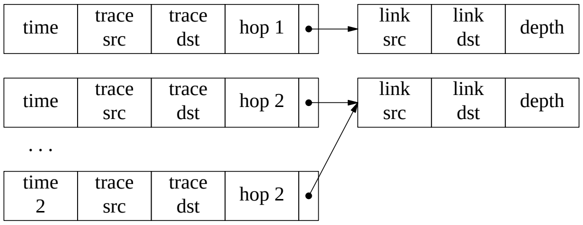
3.1 Database

The database stores tables of probing data, traceroutes, BGP updates, and data derived from these sources. This section describes the information layout (Table 4) and initial processing.

Active probes are stored in the *probes* table, which records the source and destination of each probe, and four timestamps. The timestamps capture the times at which the probe was sent, received at the destination, echoed by the destination, and the echo received at the prober.

From the probes, we identify path failures (stored in the **failures** table) and times when measurement hosts failed or rebooted (the **downtimes** table). A failure is any period of three or more *consecutive* probes between one sender and one receiver that were transmitted, but never received. We record failures longer than two minutes. Each failure event contains the source and destination of the probes, the number of consecutive lost probes, and the timestamps of the last successful probe, first lost probe, last lost probe, and next successful probe. The analysis disregards times when hosts were down. A host is down if it fails to transmit *any* probes for 15 seconds or longer. We also manually excluded some times when a host was active but had its network interface unplugged.

We take daily snapshots of our testbed topology by performing traceroutes between all pairs of monitoring hosts. We maintain this information in two tables, *topology* and *topology links*. The *topology* entry indicates the time at which the traceroute was taken and the source and destination of the traceroute. It has a link pointer that indexes into the *topology links* table to identify each hop in the traceroute. These entries also indicate the estimated "network depth" of the link (we describe how the depth is estimated in Section 3.2):



Each hop from a loss-triggered traceroute is logged in the *failure traceroutes* table. This table is identical to the *topology* table, with an additional annotation indicating which link was the last successfully probed link from the traceroute. Thus, for any failure in the **failures** table, we can easily determine the link at which the failure appeared.

To observe multiple path failures occurring at the same router, the **topology routers** table contains a mapping from each address to a canonical router ID, and the AS to which we've assigned that address. To facilitate alias resolution (Section 3.3), we also note whether the router responds with the IP ID field set in its ping replies, and we store its DNS name. The **router info** table tells us to which AS each canonical router belongs, and whether or not it is at the border of its AS.

BGP messages are recorded in the *updates* table. Each entry contains a timestamp, the prefix, whether the message was an advertisement or withdrawal, the source and destination AS, and the BGP message attributes (AS path, next-hop IP address, multi-exit discriminator, local preference, and origin type). For space efficiency, we store the AS path as a pointer into a separate table, which contains about 300,000 distinct AS paths. The *resets* table tracks when the collection host lost connectivity to its border router (e.g., power failures, reboots, network failures, etc.), thus resetting the BGP session.[1] We maintain separate update and resets tables for each BGP collection host in our testbed.

To facilitate analysis, the **failure details** table links failures to their closest traceroute. It also tracks the number of relevant BGP advertisements and withdrawals received within a one-hour window around the failure, and how soon before and after the failure we observed the first BGP message.

[1] Zebra's "debug bgp events" command logs BGP session resets. These logs show resets between our collection host and the border router, not resets between the border router and its upstream routers. More sophisticated techniques [16] now exist that may be able to identify upstream resets.

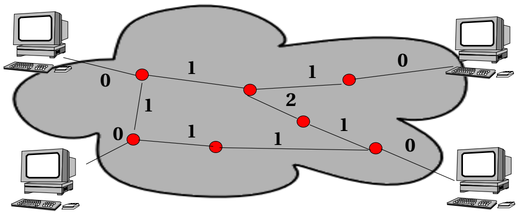


Figure 3: Estimating each link's network depth.

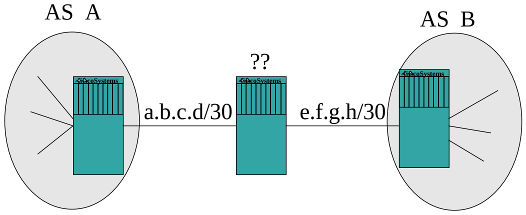


Figure 4: The AS assignment problem. Because the IP addresses on either end of an inter-AS link may be assigned from either AS A's or AS B's address space, it is difficult to tell where the boundary of a network is, and whether the center router belongs to AS A or AS B.

Our analysis characterizes failures by combining the results of the active probing and traceroute data. The **failures** table, which provides information about the duration, source, and destination of one-way failures, and the *failure traceroutes* table together (with tables that help us map IP addresses to topology information) can answer questions like "What is the average duration of a failure that appears on a network boundary?" One can also ask about the incidence of failures on interfaces, in AS's, etc. By joining failure data with our *updates* tables, we can determine the correlation between end-to-end failures and BGP instability. By characterizing failures by network depth, whether the failure appears on a network boundary, and the duration of the failure, we can investigate which types of failures tend to correlate well with BGP instability.

3.2 Network depth estimation

We are interested in whether a traceroute probe fails near an end host (either the source or destination) or in the middle of the network. Counting traceroute hops is not sufficient: while it indicates the number of hops from the source, it provides no indication about how far from the destination the failure appeared.

One previous study defines a "near source" failure as one where either the traceroute fails in the same subnet as the source host, or where the source host cannot communicate with more than one other host. "Near destination" failures are defined analogously [5]. Our approach instead uses knowledge about the testbed topology to make more general statements about how deeply in the network a particular path fails.

We assign an estimated network depth to each link in our topology based on its connectivity to other network nodes. If a router is directly connected to one of our measurement hosts, the edge between the router and the measurement node has a network depth of zero. Any edge that connects that router to other routers has a network depth of 1, and so on. If an edge can receive more than one value, we assign the smallest possible depth. Figure 3 shows an example outcome of our depth estimation algorithm. By computing the depth of all links in our *topology links* table, we can estimate the depths at which traceroutes fail.

Step	*Technique*	*Assigned*
1	Vote based on interfaces per router	2%
2	Assign interior routers to an AS	31%
3	Assign border routers based on voting by link	64%
4	Traceroute to unassigned interfaces, repeat	3%

Table 5: Iterative algorithm for assigning routers to AS's, and the percentage of 4,386 routers that were assigned by each step. The algorithm is order dependent, since steps 2 and 3 each rely on having routers already assigned to AS's.

This depth estimation algorithm has a few relevant limitations. First, the topology of the network may change from when we assigned the network depth. Second, because the algorithm assigns depths to links based on their distance from measurement hosts, the algorithm depends on the fact that none of our measurement hosts are located in the network core.[2] By validating our depth assignments against the DNS names of router interfaces, we found that our metric is reasonably good at differentiating core routers and edge routers.

3.3 Alias resolution and router placement

Observing failure locations requires an accurate AS-level topology; this requires (1) mapping interfaces to routers and (2) assigning routers to AS's. Each interface on a router has its own IP address. Assigning those IP addresses to a single router is termed *alias resolution*. We use our implementation of Rocketfuel's "Ally" technique [22] to determine whether two IP addresses are on the same router. A pair of IP addresses is a candidate for alias resolution if they both have the same next or previous hop in a traceroute (i.e., they're the hops immediately before the paths converge). For each candidate pair, we perform the alias resolution test 100 times to achieve some level of confidence that two IP addresses are on the same router. If the test is positive 80 or more times, we assign the two IP addresses to the canonical ID for that router.

Assigning IP addresses to routers does not automatically assign the routers to autonomous systems. Consider the three routers in Figure 4. The leftmost router contains IP addresses solely from AS A's address space, and the rightmost router contains IP addresses solely from AS B's address space. However, the router in the center of the figure may have interfaces with IP addresses from both AS's. Complicating matters, the endpoints of the link between AS A and AS B can come from either A or B's address space. It is difficult to determine whether the router in the center of the figure belongs to AS A or AS B.

To solve this problem, we run a voting algorithm, summarized in Table 5, on our router topology. First, for routers for which we know three or more IP addresses, we assign the router to the AS to which the majority of the addresses belong. If the interfaces belong to different autonomous systems, we also designate the router as a border router. For routers that we cannot identify this way, because we lack sufficient traceroutes across them or there is no clear majority, we assign by neighbor router votes. If the majority of the links from a router lead into one autonomous system, we assign the router to that AS. As before, if a router

[2] We have recently added hosts in the GBLX core (Table 2). Our network depth results do not include paths involving these hosts.

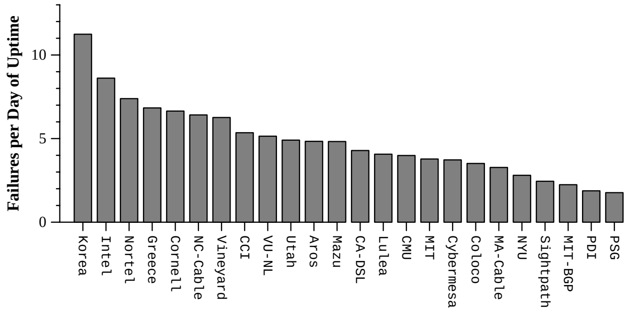


Figure 5: Failures per day of uptime by measurement host. (Seven recently added sites have very few failures to date, and thus are not shown.)

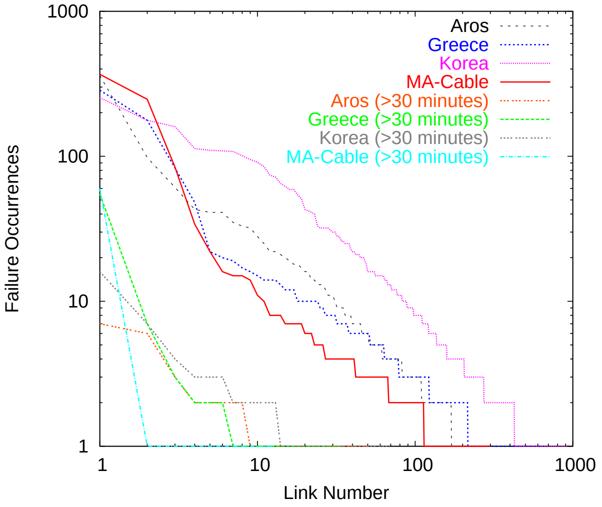


Figure 6: Failures appear in many different places, regardless of failure duration.

has links leading to multiple other systems, we also designate it as a border router. For the remaining routers, we manually initiate more traceroutes to discover either more interfaces or more neighbors and repeat the assignment process.

4. Failure characterization

In this section, we provide answers to the questions that we posed in the introduction that deal directly with the properties of failures; for example, where do failures appear and how long do they last? Next, we emulate a reactive routing system to determine under what circumstances it is effective. The results we present are based on measurement probes, traceroutes, and BGP data collected from the time we started collecting at each node until March 13, 2003, except where noted.

4.1 Failures appear often and everywhere

The paths to some hosts are much more failure-prone than others. Figure 5 shows the average number of failures per day involving each measurement host. It is very common for a host to experience multiple failures per day to at least some part of the Internet. Figure 6 shows which links are most involved in failures for four measurement hosts. For each line, the links are sorted according to the number of failures in which they appear. For example, one link was involved in failures to and from `Korea` about 2,500 times, and a different link was involved in failures to and from `Greece` about 800 times. While a small number of links are responsible for a large number of failures involving a particular host, over the course of several months failures along the paths appear in a wide variety of locations, not just one or two bad links. This phenomenon also holds for failures longer than 30 minutes. Thus, avoiding failures (or even just extremely long failures) requires more than simply avoiding a small number of bad links.

Depth	*Duration*	*Occurrences*	*Multiconnected Paths*
0-2	—	9852	4396
	2-5 minutes	7102 (72%)	3226 (73%)
	5-15 minutes	1922 (20%)	828 (19%)
	15-30 minutes	377 (4%)	162 (4%)
	>30 minutes	451 (4%)	180 (4%)
3-5	—	8182	2004
	2-5 minutes	5814 (71%)	1444 (72%)
	5-15 minutes	1620 (20%)	383 (19%)
	15-30 minutes	340 (4%)	84 (4%)
	>30 minutes	408 (5%)	93 (5%)
6+	—	175	69
	2-5 minutes	106 (61%)	42 (61%)
	5-15 minutes	42 (24%)	20 (29%)
	15-30 minutes	14 (8%)	1 (1%)
	>30 minutes	13 (7%)	6 (9%)

Table 6: Failure distribution by network depth and duration.

Type	*Duration*	*Occurrences*	*Median Duration (s)*
Intra-AS	—	11371	207.079
	2-5 minutes	8105 (71%)	—
	5-15 minutes	2219 (20%)	—
	15-30 minutes	452 (4%)	—
	>30 minutes	595 (5%)	—
Inter-AS	—	6838	199.353
	2-5 minutes	4917 (72%)	—
	5-15 minutes	1365 (20%)	—
	15-30 minutes	279 (4%)	—
	>30 minutes	277 (4%)	—

Table 7: Distribution of failures by intra-AS vs. on a network boundary, and by duration. In both cases, the median failure length is about 3.5 minutes, and roughly 70% of failures are less than 5 minutes. Roughly 63% of all path failures appear inside an AS.

Table 6 shows that many path failures appear within two hops of an end host (analogous to a "near-src" or "near-dst" failure in prior work [5]); this is particularly true for failures that occur on paths were both ends are multiconnected. (We call a host *multiconnected* if its LAN or campus-area network has at least two upstream providers.) If many failures occur close to end hosts, how well can reactive routing work in practice, especially if the host is not multi-homed? We address this question in Section 4.3. Failures that occur on paths with multiconnected endpoints experience the same distribution for failure duration; thus, while multiconnectedness may serve to improve robustness, it does not appear to improve failover time.

Table 7 shows that about 62% of failures appear inside an AS as opposed to on a network boundary. We initially believed that most failures would appear on boundaries, especially since 2/3 of the routers in our topology are on network boundaries. This

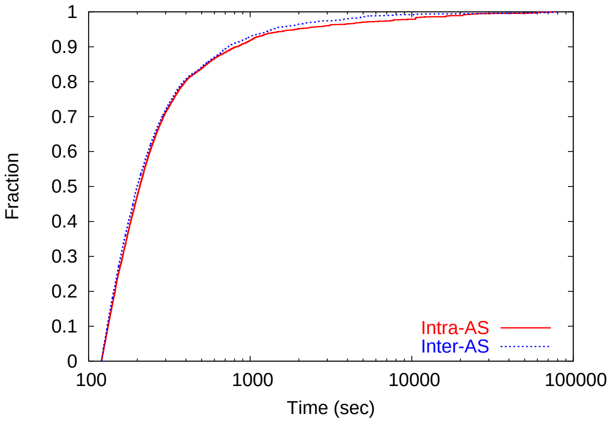


Figure 7: Roughly 90% of failures last less than 15 minutes, and about 70% of all failures last less than five minutes. Failure duration does not appear to depend on whether that failure occurs on a network edge.

	All Hosts	**BGP Hosts**		
Length (min)	*RON Success*	*RON Success*	*BGP*	*BGP Before*
6-15	52% (1767/3400)	44% (151/343)	52% (78/151)	24% (36/151)
15-30	53% (658/1240)	45% (53/117)	62% (33/53)	28% (15/53)
> 30	45% (959/2110)	46% (66/142)	67% (44/66)	21% (14/66)
TOTAL	50% (3384/6750)	45% (270/602)	57% (155/270)	24% (65/270)

Table 8: Percentage of failures of each duration that a reactive routing network can route around. Our RON emulation masks more than half of a path failure about 50% of the time. the time.

difference may arise because we are looking at *failures*, not congestion: First, we observe where failures *appear*; faults near the edge may manifest themselves in the core through routing updates. Second, faults on the network edge may be masked by alternate paths, whereas intra-AS failures may reflect fundamental forwarding problems (e.g., persistent forwarding loops, failure of a long haul link, etc.). Supporting our observation, we note that Chandra et al. found similar results in their own traceroute-based study [5]. Finally, as we will see, because the location of a failure affects whether it coincides with BGP messages, the fact that many failures appear inside AS's is an important observation.

4.2 Many failures are short

Figure 7 shows that the median duration of long ($>$ 2 minute) failures is just over three minutes, and that about 90% of failures last less than 15 minutes. The duration of failures does not depend on whether they appear at the edge of an autonomous system or inside of it. As shown in Table 6, failure duration does not appear to depend on whether that failure is closer to the network edge or the core. Given that most failures are short, reactive routing systems must react quickly to route around these failures before they end.

4.3 Multihoming improves reactive routing

As pointed out in Section 1, RONs detect Internet path failures using pairwise active probing between overlay nodes and send packets along one-hop alternate paths in the overlay that avoid failures [2]. The study found that a RON could usually bypass 30-minute Internet failures and dramatically reduce the loss rate between two hosts when the loss rate was high. However, RONs perform less well over 30-minute periods with comparatively lower loss rates. We hypothesize that these 30-minute periods with lower loss rates might in fact be one or more short failures that we could evaluate with our data.

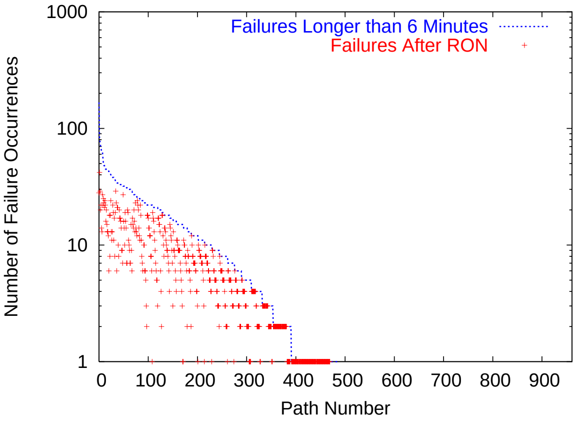


Figure 8: The number of failures longer than 6 minutes experienced by each path, and the number of failures that could not be corrected at all by a RON (50% of all failures) experienced by each path.

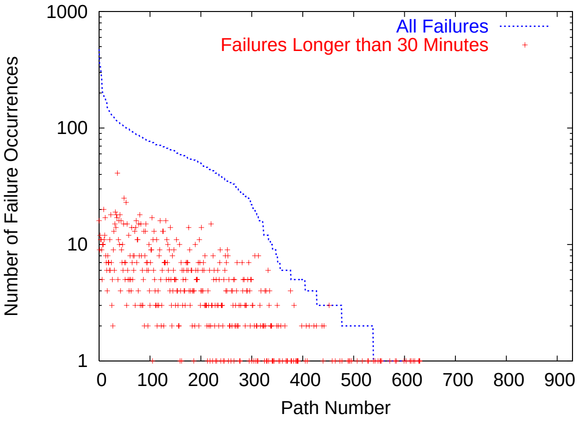


Figure 9: The success of an reactive routing network depends on which paths commonly experience failures. This figure ranks the 930 paths according to the number of failures that each path experiences, for various failure durations (approximately 300 paths did not experience failures during the collection period).

To evaluate the effectiveness of reactive routing in the presence of Internet path failures, we use our **failures** table to emulate the behavior of a RON [2]. We assume that a reactive routing system like RON that performs aggressive active probing can find a one-hop alternate path within 30 seconds of a failure, if such a path exists. Based on this assumption, we determine which failures this reactive routing system could successfully route around. Of these failures, we determine which ones were preceded by a flurry of BGP messages that could have been used as a predictor of poor path performance.

Although the limitations of our emulation prevented us from evaluating the effectiveness of a RON for failures shorter than 6 minutes, our results suggest that a RON should succeed in routing around approximately 50% of failures, with slightly more success for failures shorter than 30 minutes. We declare success if at least half of the failure duration can be avoided.

For each pair of nodes in our measurement testbed (i.e., each path in the overlay), we ask: (1) What fraction of all failures occur on that path? and (2) For each path, what fraction of those failures can a RON route around? The first question addresses how the failures we see are distributed among paths on the overlay; the second determines whether RONs can mask failures on certain paths, and whether those paths are dependent on underlying topology. We examine these characteristics for failures of various durations.

The fact that many failures occur within three hops of an end host (Table 6) caused us to speculate that a RON would have more success routing around failures between hosts that are multiconnected. Of the 25 paths (2.6% of the paths and 1.1% of all failures) where RON always masked failures, 18 (72%) were between two multiconnected hosts and all 25 had at least one multiconnected host. In contrast, of the 136 paths where RON never masked failures (14.6% of paths and 4% of failures), only 25.7% were between two multiconnected hosts and 22% did not have a multiconnected host at either end. Because failures tend to occur near end hosts, a RON with multiconnected hosts seems to have better success in routing around path failures.

The re-routability of failures also varies with duration. Figures 8 and 9 rank the 930 pairwise paths in the topology by the number of failures experienced by each one. For example, the dotted line in Figure 8 shows that one path in the topology experienced about 150 failures over the course of the trace, another path experienced about 80 failures, etc. The points show, for each path, how many failures could not be corrected by reactive routing. Figure 8 shows that reactive routing does not correct routing problems equally on all paths. Figure 9 shows that paths that fail more often are not necessarily the sites of long failures.

5. Correlating failures and BGP instability

While BGP itself is a dynamic routing protocol that reacts to various network faults and path failures, prior studies observed that convergence after a fault can take as long as 15 minutes [11]. During this convergence, BGP sends numerous routing messages. Reactive routing schemes can perhaps make use of the fact that BGP routing instability coincides with poor path performance to detect and predict path failures.

Because of BGP's slow convergence and general routing instability, a path failure typically coincides with a large number of BGP messages. BGP messages may follow the appearance of a failure as the routing system reacts to the failure. In cases where routing changes signal external events such as network maintenance, or where routing problems are the cause of failures, BGP messages may actually precede a failure.

Failures that appear on network boundaries and closer to the network core correlate well with BGP messages, and vice versa. This observation helps explain why failures involving some hosts correlate strongly with BGP messages, while others do not. Knowing this, we look at the causal relationships between BGP messages and path failures for failures between `MIT` and several testbed hosts over a 13-month period, as well as the temporal correlation between path failures and BGP messages for some of these hosts. Finally, we examine how a high level of BGP message activity can indicate and predict path failures.

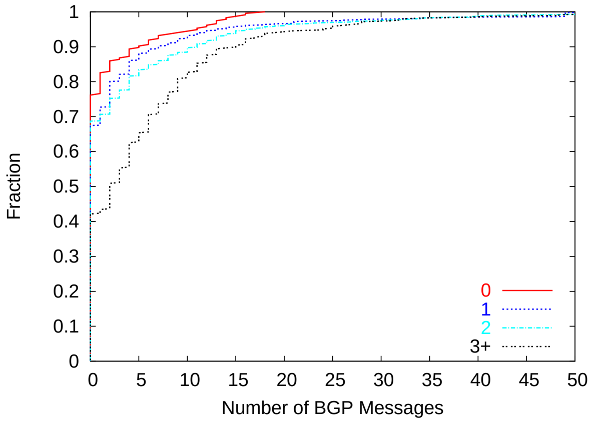


Figure 10: Failures near end hosts tend not to coincide with BGP messages.

5.1 Core failures show BGP instability

Using the **failure details** tables, we determine the fraction of failures that coincide with n or fewer BGP messages within a 60-minute time window (30 minutes preceding and 30 minutes after). We join this information with *topology links* to determine whether the amount of BGP activity that coincides with a failure correlates with the estimated depth at which the failure appeared. Figure 10 shows that failures that appear closer to an end host are much less likely to have a corresponding BGP message. Faults that occur close to end hosts are much more likely to be part of a prefix aggregate or default-routed stub AS, whereas failures that appear farther from end hosts are much more likely to coincide with routing instability that is visible from the end hosts. This observation helps explain why failures involving some hosts correlate better with BGP messages than others. We initially thought that failures that appeared on an autonomous system boundary would coincide with BGP messages, whereas intra-AS failures would not result in BGP messages visible at the network boundary, but this does not appear to be the case. Regardless of whether the failure appears on an AS boundary or not, at least one BGP message coincides with a failure 35% of the time. This is likely because BGP messages equally reflect failures at network boundaries, as well as intra-AS failures whose effects are propagated by BGP.

5.2 Failures often precede BGP instability

To measure the temporal correlation between BGP messages and end-to-end failures, we define a stochastic process for each of these events and observe how these processes are correlated in time for each host. $B(t)$ describes the relevant BGP messages *for a particular host*. A BGP message is relevant if its prefix contains the monitoring host in question, even if it is not the best matching prefix for that host. We examine a time window, w, of 100 seconds during which we consider a BGP message "recent." Thus, $B(t) = 1$ if there exists a BGP message within the time interval $[t, t + w)$, and 0 otherwise. We analogously compute the failure process, which is 1 if a failure occurred during that

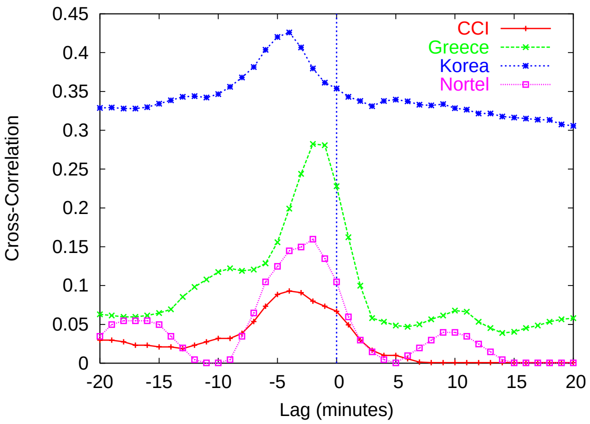


Figure 11: Cross-correlation between path failures and BGP messages. In general, path failures lead BGP messages by 2-4 minutes. However, for some hosts, failures can follow BGP messages by many minutes.

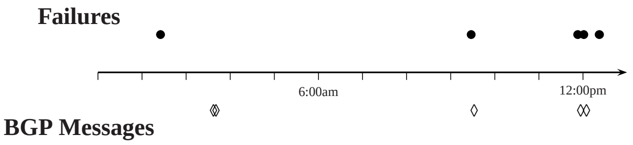


Figure 12: Example timeline for failures and BGP messages between `Korea` and `MIT` on January 29, 2003 (times are EST).

time window. We assume that these processes are wide-sense stationary.

To see how failures and BGP events are related, we examine their cross-correlation R_{xy}, which quantifies the extent to which two signals are correlated at different time offsets. Given two random processes, $x(t)$ and $y(t)$, $R_{xy}(\tau)$ reflects the correlation between the signals $x(t)$ and $y(t-\tau)$ for all possible delays τ. For example, if $y(t) = x(t-\delta)$, where δ is some constant time delay, then $R_{xy}(\tau)$ has a maximum at $\tau = -\delta$, since the signal $y(t)$ lags $x(t)$ by exactly δ time units. We use the normalized cross-correlation function:

$$R_{xy}(\tau) = \frac{E[(x(t) - \mu_x)(y(t-\tau) - \mu_y)]}{\sigma_x \sigma_y}$$

We consider BGP messages and path failures as two signals indexed in time and observe the cross-correlation between these signals to determine the time relationships between them. As expected, failures most commonly precede BGP messages by about 2-4 minutes, as shown in Figure 11. This appears to be the common case of a BGP announcement following a failure due to slow convergence (a phenomenon observed by Labovitz et al. [11]). There is also a fair amount of correlation for both more positive and more negative lags. BGP messages sometimes trail failures by 15 minutes or longer, but it is also common for BGP messages to precede failures.

5.3 BGP instability can also precede failures

If a path failure occurs to or from a host at a certain time, what is the probability that a relevant BGP message (i.e., a BGP message for a prefix that contains that host) will appear within a certain amount of time before or after that path failure event? For example, Figure 12 shows the path failures between `MIT` and `Korea` and the BGP messages heard at MIT for prefixes containing `Korea` for 12 hours on January 29, 2003; for any particular failure (black circle), what is the probability that we will see a BGP message (diamond) within a certain time before or after that failure? In this section, we are not trying to answer the question of whether BGP messages actually *cause* failures, or vice versa. Rather, we try to understand what the occurrence of one event tells us about the occurrence of another.

We originally thought that BGP messages would almost always coincide with path failures and would occur several minutes after the occurrence of the path failure, in accordance with prior studies [11]. To examine this, we correlated BGP messages heard at our MIT BGP collection host with path failures involving `MIT` and another host.

To better understand the causal relationships between BGP messages and end-to-end failures, we computed the conditional probability of seeing a failure within a certain time before and after a BGP message. When a BGP message occurs for a specific prefix, how likely is it that we will see a failure to our monitoring host contained in that prefix, and when can we expect to see such a failure?

Given a BGP message at time 0, we examine the conditional probability of having seen a failure within t seconds *before* the message, and the probability of seeing a failure within t seconds *after* the message:

$$\begin{aligned} f_{\text{failure,pre}}(t) &= P(\text{failure in } (-t,0] | \text{BGP at time } 0) \\ f_{\text{failure,post}}(t) &= P(\text{failure in } [0,t) | \text{BGP at time } 0) \end{aligned}$$

We examine only the failure closest to time 0 in each direction over a window 15 minutes before and after every BGP message.[3] For any given host, $f_{\text{failure,post}}(t)$ is the number of times a failure was observed between $[0,t)$ divided by the total number of BGP messages seen by that host; analogously for the *pre* case.

Conversely, we investigate how often an end-to-end failure to or from a particular host is preceded or followed by a BGP message for a prefix that contains that host's IP address. To do so, we followed a similar methodology to find $f_{\text{bgp,post}}(t)$, the conditional probability of seeing a BGP message for a host in time $[0,t)$, given a failure at time $t = 0$ and no BGP messages for that host in $[0,t)$, as well as $f_{\text{bgp,pre}}(t)$, the conditional probability of seeing a BGP message for a host in time $(-t,0]$, given a failure at time $t = 0$ and no BGP messages for that host in $(-t,0]$.

Figure 13 shows the probability of hearing a relevant BGP announcement at a certain time before or after the occurrence of a failure up to 15 minutes on either side of a path failure. We show the results between `MIT` and 8 representative hosts, omitting the rest for clarity. For example, failures to and from `Korea` are followed by a BGP announcement about 65% of the time and are preceded by a BGP message about 50% of the time. On the other hand, failures between `MIT` and `CA-DSL` are rarely preceded by BGP messages and are followed by BGP messages within 15 minutes only 8% of the time.

[3] Experience with longer time windows showed that the 15-minute window captures most relevant failures.

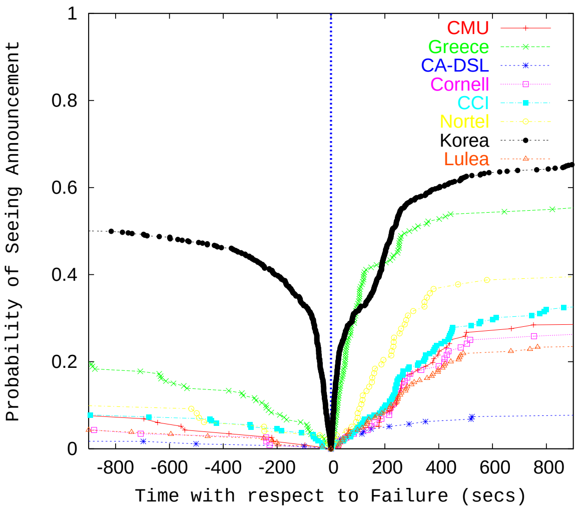


Figure 13: Given a path failure at time zero between MIT and some other host, what is the probability we will hear a BGP message at MIT for that host within a certain period of time?

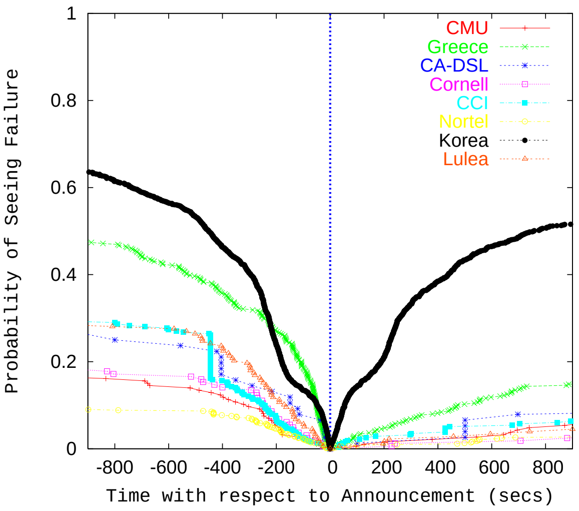


Figure 14: Given a BGP message for a host, what is the probability that a path failure will occur between MIT and that host within a certain period of time?

We were originally surprised by the wide disparity in these correlations. However, the results from the Section 5.1 help to explain these observations. We were also surprised by BGP messages preceding path failure events. We believe that these messages could arise in a few ways. Heavy congestion could conceivably coincide with an increase in routing traffic and precede a path failure. Second, a single route change could cause a cascade of updates during delayed convergence that cause the route to be suppressed due to route flap damping [14], resulting in a path failure from the perspective of the end host. Finally, route changes may be indicative of ongoing external events, such as configuration changes, that have an effect on network reachability.

In Figure 14, we examine the converse question: given a BGP message for a particular host, what is the probability of a path failure occurring within a certain period of time before or after message? While general correlation trends are similar to that in Figure 13, one interesting difference is CA-DSL, for which path failures are rarely followed by BGP messages, but path failures commonly precede BGP messages. For this host, many failures occur without routing instability, but routing instability often indicates the presence of a failure in the preceding 15 minutes. In this case, BGP messages are not a good predictor of path failures. On the other hand, for sites such as Korea and Greece failures follow BGP messages within 15 minutes 50% and 15% of the time, respectively; in cases such as these, BGP messages may be able to predict some path failures; we explore this idea further in Section 5.4.

5.4 BGP can sometimes predict failures

Because path failures can coincide with BGP message activity, can we detect the presence of such path failures, knowing only BGP message activity? Such a method could potentially allow reactive routing systems to passively discover failures along certain paths with a much lower rate of active probes, which is increasingly important as the network grows larger.

How well does an increase in BGP message traffic indicate the occurrence of a failure? Intuition suggests that many BGP messages within a small period of time might indicate the occurrence of a route withdrawal, as intermediate routers explore alternate paths.

To answer this, we derive a decision rule for determining if a failure event F occurred, given n BGP messages for a host within a time window of 15 minutes.

Let b be a random variable that is the number of relevant BGP messages heard for a source over a 15-minute time window, and let F indicate the observation of an end-to-end path failure during that time period. We first compute the conditional densities $f_{b|F}(b|F)$ and $f_{b|\overline{F}}(b|\overline{F})$ for each host, which tell us how n is distributed under each hypothesis.

We define our decision rule in terms of the likelihood ratio

$$\Lambda(r) \quad = \quad \frac{f_{b|F}(b=n|F)}{f_{b|\overline{F}}(b=n|\overline{F})}$$

where n is the number of observed failures within a 15-minute time window. If $\Lambda(r) > \gamma$, we say that a failure has occurred.

Given $\Lambda(r)$, we determine the probability of a false positive, P_{FP}, and the probability of detection, P_D, for various values of γ. This is commonly called the receiver operating characteristic, or ROC, for a decision rule [8].

BGP-based prediction achieves a 0.5 probability of detection for false positive rates of less than 0.01 for many hosts, as shown in Figure 15. Each line on this graph corresponds to failure tests between MIT and some other host; each point on a line corresponds to the *threshold* number of BGP messages within that 30 minute time window for declaring that a failure event occurred. For example, as shown in Figure 15, setting the threshold to 1 in the case of Nortel gives a probability of detection of 0.5 with a false positive rate of about 0.001. Previous work defined failure predictors based on observations of past round-trip times [3]. In general, their best predictor for path failures (termed “level 6 degradations”) still does not have much predictive value—a 0.6

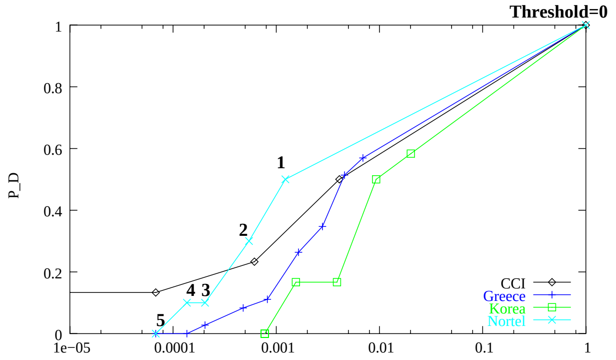


Figure 15: Receiver operating characteristics using a 30-minute advertisement window of announcement observation. Setting the threshold for declaring a path failure results in different detection and false alarm (or false positive) probabilities.

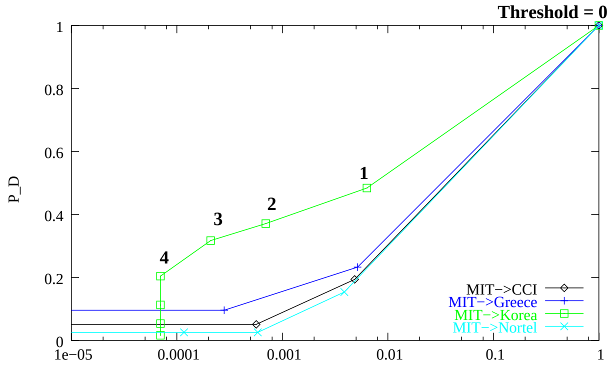


Figure 16: Receiver operating characteristics using a 30-minute advertisement window of *failure* observation. For paths where many failures happen in a short time span, failures can be a good predictor of future failures.

false positive probability for a 0.5 probability of detection in the best case.

While path failures between `MIT` and `Korea` often coincide with BGP messages, the probability of detection is rather low (around 0.2) when the threshold is set at one message. This occurs because many of `Korea`'s path failures fall into the same 30-minute failure window and are counted as only one distinct event (one example of this is shown in Figure 12). Many of the time windows when a path failure occurs have no announcements, and vice versa. For this reason, *failures themselves* are generally a better predictor of failures on the path between `MIT` and `Korea`. Figure 16 shows that using failures to predict future failures on the path between `MIT` and `Korea` achieves a much lower false positive rate for a given probability of detection. In general, however, BGP messages provide a higher probability of detection for a given false positive rate.

5.5 Can BGP improve reactive routing?

We ask two questions about the failures that a reactive system can route around: (1) Did a BGP message precede the failure? and (2) Did a BGP message occur within the timespan of the failure? The first question tries to figure out what fraction of failures reactive routing can *pre-emptively* route around by using BGP messages as a predictive indicator. The second considers the fraction of path failures that could have been detected and routed around with fewer active probes.

Using the reactive routing emulation process described in Section 4.3 in combination with the *updates* tables, we considered whether BGP messages could help reactive routing schemes predict or detect failures. We examined the success for routing around failures at hosts for which we also had a BGP feed. Table 8 shows that many maskable failures experienced a BGP message during the failure, suggesting the potential for failure detection with less active probing. Table 8 also shows that more than $1/5$ were preceded by a BGP announcement within 30 minutes of the failure; these failures may be avoidable by using BGP as a warning indicator. Conversely, reactive routing masked about 25% of failures that did not coincide with any BGP messages; this suggests that reactive routing can improve the performance of many failures that are not even visible at the interdomain routing layer.

6. Related Work

A large body of work has examined individual components of the interactions we study in this paper. We believe that our work is the first to provide an integrated analysis of end-to-end ping data, traceroute information, and BGP routing traffic. In addition, we explore the implications that our results have for reactive routing systems.

Many studies have looked at individual components of the interactions we study in this paper, examining end-to-end Internet reachability, routing problems, topology construction, and reactive routing.

Paxson studied Internet failures and routing pathologies by running periodic traceroutes between 37 distributed sites on the Internet [18]. His study noted that "major routing pathologies" disrupt two-way connectivity between hosts between 1.5% to 3.4% of the time. These traceroutes provided a "host's-eye" view of routing loops and failures, but did not distinguish one-way outages or observe BGP routing traffic. Similarly, the Skitter project uses traceroutes to map the Internet topology and identify critical paths [4].

Labovitz *et al.* examined BGP routing failures between various Internet Service Providers [12], finding that only 40% of routing failures are repaired within 10 minutes. Paxson's data and our data show that end-to-end path failures are repaired much more quickly; in our dataset, more than 80% of path failures are repaired within 10 minutes. Our analysis shows that many end-to-end failures are not reflected at all in BGP traffic.

Chandra *et al.* model two-way path failures using traceroute and HTTP probes [5]. They note that long failures account for a significant fraction of total failure duration. Similarly, 10% of our observed failures last longer than 15 minutes. The authors note that their HTTP and traceroute probes underestimate failures near the destination. The use of correlated active probes and traceroutes eliminates this bias. The network depth estimate in Section 3.2 builds on their near-source failure location technique. Both their study and ours find that failures frequently occur within a stub AS or inside an autonomous system.

For bypassing end-to-end outages in a RON-like manner [2], it appears that more complex Markov-model path predictors offer only a slight improvement over the simple weighted average prediction used in RON [3]. Our work suggests, however, that using BGP routing instability information can provide a larger improvement for reactive routing systems.

Zhang *et al.* note that routing stability and other path properties vary widely with network location [25]. Similarly, we find that the correlation between routing instability and path failures varies widely with both the hosts between which the failure occurs, and the location in the network where it appears.

Freedman notes that BGP instability often precedes end-to-end congestion and speculates on its use as a predictor of path congestion [9]; analogously, we explore how this information can be used to react to outages. Others note that due to BGP's slow convergence and path exploration, as well as the likelihood of BGP session resets on congested links, a single route withdrawal can create a considerable amount of BGP traffic during times of path failure or persistent congestion [11, 23].

7. Conclusion

In this paper, we provide answers to three questions about Internet path failures: (1) Where are failures likely to appear? (2) How long do failures last? and (3) How do failures correlate with routing protocol messages? While a few locations experience failures on a regular basis, the appearance of failures is not isolated to small set of locations but seems to be a general property of Internet paths. We find that failures are more likely to appear within an autonomous system than on the boundary, thus making reactive routing techniques particularly important. 70% of the failures we observe last less than 5 minutes, and 90% are shorter than 15 minutes. Path failures that appear closer to the network core are more likely to coincide with BGP messages for the path's destination than failures that appear near end hosts. When they coincide, failures typically precede BGP messages by about 4 minutes, but it is not uncommon to see BGP messages for a destination both before and after a path failure involving that destination. In these cases, BGP messages can be used to both predict and detect failures.

We examine the effects of path failures and related BGP messages on reactive routing techniques, such as RON, and find that overlay networks can typically route around 50% of failures, independent of failure duration. Overlay networks seem to be more effective at routing around failures between hosts that have at least one "large" AS along the path. Further investigation of this is an avenue for future work. We find that a path that experiences relatively many short failures does not necessarily experience many long failures. That is, "bad paths" for short failures may be different for "bad paths" for long failures. Finally, more than 20% of the path failures that an overlay network was able to mask were preceded by at least one BGP message, suggesting that reactive routing schemes can be improved by using BGP instability as a leading indicator of path failures.

Acknowledgments

The NMS and PDOS researchers at MIT and our reviewers provided valuable feedback on this work. We are very grateful to the people and institutions hosting the RON testbed, listed at `http://nms.lcs.mit.edu/ron/thanks.html`. We owe particular thanks to people who provided BGP feeds: Andre Barrette, Eric Bates, Mike Biesele, Randy Bush, James Kretchmar, Matt Meyer, Jeffrey Papen, and Josh Richards.

This research was sponsored by the Defense Advanced Research Projects Agency (DARPA) and the Space and Naval Warfare Systems Center, San Diego, under contract N66001-00-1-8933.

8. References

[1] Amini, L., Shaikh, A., and Schulzrinne, H. Issues with inferring Internet topological attributes. In *Proc. SPIE ITCOM* (Boston, MA, August 2002), vol. 4685, pp. 80–90.

[2] Andersen, D. G., Balakrishnan, H., Kaashoek, M. F., and Morris, R. Resilient Overlay Networks. In *Proc. 18th ACM SOSP* (Banff, Canada, Oct. 2001), pp. 131–145.

[3] Bremler-Barr, A., Cohen, E., Kaplan, H., and Mansour, Y. Predicting and bypassing end-to-end Internet service degradations. In *Proc. ACM SIGCOMM Internet Measurement Workshop* (Marseille, France, November 2002).

[4] CAIDA's Skitter project, 2002. `http://www.caida.org/tools/measurement/skitter/`.

[5] Chandra, B., Dahlin, M., Gao, L., and Nayate, A. End-to-end WAN Service Availability. In *Proc. 3rd USITS* (San Francisco, CA, 2001), pp. 97–108.

[6] Chang, D.-F., Govindan, R., and Heidemann, J. An empirical study of router response to large BGP routing table load. Tech. Rep. ISI-TR-2001-552, USC/Information Sciences Institute, December 2001.

[7] Donelan, S. Update: CSX train derailment. `http://www.merit.edu/mail.archives/nanog/2001-07/msg00351.html`.

[8] Egan, J. *Signal Detection Theory and ROC Analysis*. Academic Press, New York, 1975.

[9] Freedman, A. Active UDP and TCP performance during BGP update activity. In *Proc. Internet Statistics Metrics and Analysis Workshop* (Leiden, The Netherlands, October 2002). `http://www.caida.org/outreach/isma/0210/ISMAagenda.xml`.

[10] Gao, L. On inferring automonous system relationships in the Internet. *IEEE/ACM Transactions on Networking 9*, 6 (December 2001), 733–745.

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

[12] Labovitz, C., Ahuja, A., and Jahanian, F. Experimental Study of Internet Stability and Wide-Area Backbone Failures. In *Proc. 29th International Symposium on Fault-Tolerant Computing* (June 1999).

[13] Mahajan, R., Wetherall, D., and Anderson, T. Understanding BGP misconfiguration. In *Proc. ACM SIGCOMM* (Aug. 2002). (to appear) `http://www.cs.washington.edu/homes/ratul/bgp/bgp-misconfigs.ps`.

[14] Mao, Z. M., Govindan, R., Varghese, G., and Katz, R. Route Flap Damping Exacerbates Internet Routing Convergence. In *Prof. ACM SIGCOMM 2002* (Pittsburgh, PA, August 2002).

[15] Miller, G. Overlay routing networks (akarouting), Apr. 2002.

[16] Nichol, D. Detecting behavior propagation in BGP trace data. In *Proc. Internet Statistics Metrics and Analysis Workshop* (Leiden, The Netherlands, October 2002). `http://www.caida.org/outreach/isma/0210/talks/david.pdf`.

[17] Opnix. Orbit: Routing Intelligence System. `http://www.opnix.com/newsroom/OrbitWhitePaper_July_2001.pdf`, 2002.

[18] Paxson, V. End-to-End Routing Behavior in the Internet. *IEEE/ACM Transactions on Networking 5*, 5 (1997), 601–615.

[19] MIT RON Project. `http://nms.lcs.mit.edu/ron/`.

[20] RouteScience. `http://www.routescience.com/`.

[21] Sockeye. `http://www.sockeye.com/`.

[22] Spring, N., Mahajan, R., and Wetherall, D. Measuring ISP topologies with Rocketfuel. In *Proc. ACM SIGCOMM* (Aug. 2002).

[23] Wang, L., et al. Observation and analysis of BGP behavior under stress. In *Proc. ACM SIGCOMM Internet Measurement Workshop* (Marseille, France, November 2002).

[24] Gnu Zebra. `http://www.zebra.org/`.

[25] Zhang, Y., Duffield, N., Paxson, V., and Shenker, S. On the constancy of Internet path properties. In *Proc. ACM SIGCOMM Internet Measurement Workshop* (San Francisco, CA, November 2001).