Studying Black Holes in the Internet with Hubble

Ethan Katz-Bassett[*] *Harsha V. Madhyastha*[*] *John P. John*[*] *Arvind Krishnamurthy*[*]
David Wetherall[†] *Thomas Anderson*[*]

Abstract

We present *Hubble*, a system that operates continuously to find Internet reachability problems in which routes exist to a destination but packets are unable to reach the destination. *Hubble* monitors at a 15 minute granularity the data-path to prefixes that cover 89% of the Internet's edge address space. Key enabling techniques include a hybrid passive/active monitoring approach and the synthesis of multiple information sources that include historical data.

With these techniques, we estimate that *Hubble* discovers 85% of the reachability problems that would be found with a pervasive probing approach, while issuing only 5.5% as many probes. We also present the results of a three week study conducted with *Hubble*. We find that the extent of reachability problems, both in number and duration, is much greater than we expected, with problems persisting for hours and even days, and many of the problems do not correlate with BGP updates. In many cases, a multi-homed AS is reachable through one provider, but probes through another terminate; using spoofed packets, we isolated the direction of failure in 84% of cases we analyzed and found all problems to be exclusively on the forward path from the provider to the destination. A snapshot of the problems *Hubble* is currently monitoring can be found at `http://hubble.cs.washington.edu`.

1 Introduction

Global reachability – when every address is reachable from every other address – is the most basic goal of the Internet. It was specified as a top priority in the original design of the Internet protocols, ahead of high performance or good quality of service, with the philosophy that "there is only one failure, and it is complete partition" [4]. Today, middleboxes such as NATs complicate this picture by artificially restricting connectivity to addresses within some customer networks. Yet within the default-free core of the Internet, it should be the case that if there is a working physical path that is policy-compliant, then there will be a valid BGP path, and if there is a valid BGP path, then traffic will reach the destination. However, this is not always the case in practice; traffic may disappear into *black holes* and consistently fail to reach the destination. Outages that are not transient are problematic, as an operator generally has little visibility into other ASes to discern the nature of an outage and little ability to check if the problem exists from other vantage points. For example, black holes are a recurring theme on the Outages [28] and NANOG [29] mailing lists [7], with users asking whether others can reach their prefixes, or posting when they are unable to reach certain destinations, to ask if others see the same problem or know the cause.

Internet operations would thus benefit from having a system that automatically detects reachability problems and aids operators in locating the network entity (an AS, a router) responsible for the problem. Previous systems addressed aspects of this goal in different ways. A number of systems monitored reachability status in real-time, but within contexts that are narrower than the whole Internet, such as a testbed [23, 1, 6, 12], an autonomous system [34, 24], or a particular distributed system's clients [35]. Other systems such as *iPlane* [20] have broad and continuous Internet coverage but, being designed for other purposes, monitor at too infrequent a rate to provide real-time fault diagnosis. Still another body of work detects certain reachability issues in real-time at the Internet scale by passively monitoring BGP feeds [8, 34, 3]. But these techniques isolate anomalies at the level of autonomous systems and are thus too coarse-grained from the perspective of a network operator. More importantly, as our data will show, relying on BGP feeds alone is insufficient because the existence of a route does not imply reachability; BGP acts as a control plane to establish routes for the data plane on which Internet traffic flows, and connectivity problems that do not present themselves as events on the monitored control plane will evade such systems.

Our goal is to construct a system that can identify Internet reachability problems over the global Internet in real-time and locate the likely sources of problems. We call our system *Hubble* and focus initially on reachability problems, though we believe that the principles of our system extend to other data-plane problems. The major challenge in building *Hubble* is that of scale: how can we provide spatial and temporal coverage that scales to the global Internet, monitoring the data-plane from all vantages to all destinations, without requiring a prohibitive number of measurement probes.

In this paper, we describe the design and evaluation of *Hubble*. To identify potential problems, the system mon-

[*]Dept. of Computer Science, Univ. of Washington, Seattle.
[†]Univ. of Washington and Intel Research.

itors BGP feeds and continuously pings prefixes across the Internet from distributed PlanetLab sites. It then uses traceroutes and other probes from the sites to collect details about identified problems and uses a central repository of current and historical data to pinpoint where packets are being lost. We show that it monitors 89% of the Internet's edge prefix address space with a 15-minute time granularity and discovers 85% of the reachability issues that would be identified by a pervasive probing approach (in which all vantage points traceroute those same prefixes every 15 minutes), while issuing only 5.5% as many probes. We believe *Hubble* to be the first real-time tool to identify reachability problems on this scale across the Internet. The next largest system of which we are aware, PlanetSeer, covered half as many ASes in a 3-month study and monitored paths only for the small fraction of time that they were in use by clients [35]. *Hubble* has been running continuously since mid-September, 2007, identifying over 640,000 black holes and reachability problems in its first 4 months of operation. It also performs analysis using fine-grained real-time and historical probes to classify most problems, a step towards diagnosis for operators.

Hubble relies on two high-level techniques:

Hybrid monitoring: *Hubble* uses a hybrid passive/active monitoring approach to intelligently identify target prefixes likely to be experiencing problems. The approach combines passive monitoring of BGP feeds for the entire Internet with active monitoring of most of the Internet's edge. The two monitoring subsystems trigger distributed active probes when they suspect problems. Currently, they identify targets that might not be globally reachable, but in the future we plan to also look at performance issues such as latency. The hybrid approach allows *Hubble* to monitor the entire Internet while still providing router-level probe information from diverse vantage points at a reasonably fast pace during problems.

Synthesis of multiple information sources: In order to provide as much detail on problems as possible, *Hubble* combines multiple sources of information. For example, *Hubble* maintains historical records of successful traceroutes from its vantage points to destinations across the Internet and monitors the liveness of routers along these routes. When it finds that one of its vantage points is unable to reach a destination, it compares current router-level probe data from that site to its historical data and to probes from other sites, to determine the extent and possible location of the problem.

We also present observations on Internet reachability made from three weeks of *Hubble* data. We found reachability problems to be more common, widespread and longer lasting than we had expected. Over three weeks, we identified more than 31,000 reachabilility problems involving more than 10,000 distinct prefixes. While many problems resolved within one hour, 10% persisted for more than one day. Many of the problems involved partial reachability, in which some vantage points can reach a prefix while others cannot, even though a working physical path demonstrably exists. This included cases in which destination prefixes with multi-homed origin ASes were reachable through one of the origin's providers and not another. It suggests that edge networks do not always get fault tolerance through multi-homing. Finally, we observed that many Internet reachability problems were not visible as events at commonly used BGP monitors. That is, BGP monitoring alone is not sufficient to discover the majority of problems.

The rest of this paper is organized as follows. We define the reachability problem in Section 2. In Section 3, we describe the design of *Hubble*. We present an evaluation of *Hubble* in Section 4 and use it to study Internet reachability in Section 5. Related work is given in Section 6 and we conclude in Section 7.

2 Problem

In this section, we present necessary background on Internet routing, then define the reachability problems we study.

2.1 Background

An autonomous system (AS) on the Internet is a collection of routers and networks that presents a common routing view to the rest of the Internet. Routes on the Internet are determined on a per-prefix basis, with the prefix comprising all IP addresses with p as their first n bits typically written as p/n. ASes exchange routes using the routing protocol BGP, with an AS announcing to its neighbor its ability to route to a particular prefix by giving the AS path it plans to use. An origin AS for a prefix is the first AS on an AS path to that prefix, and a multi-homed AS is one with multiple provider ASes.

2.2 Defining reachability problems

We are interested in reachability problems with four characteristics:

- *Routable prefix.* We ignore cases in which we do not expect the prefix to be reachable, either because the prefix has never been reachable or the prefix has been completely withdrawn from BGP tables. BGP monitoring easily detects these problems already.
- *Persistent.* Although we may happen to detect short term route failures, such as those experienced during BGP convergence [18, 33, 16], our focus is on detecting persistent issues. We consider only problems that persist through 2 rounds of quarter-hourly probes.
- *Not simply end-system or end-network failures.* We are primarily interested in problems in which the Internet's routing infrastructure fails to provide connec-

tivity along advertised AS paths, rather than problems in which the traffic traverses the AS path, but the specific destination or prefix happens to be down. As such, though we also track whether probes reach the destination or its prefix, we make judgments on reachability problems based on connectivity to the origin AS.

- *Not simply source problems.* We are concerned with the reachability of destinations on an Internet-scale, rather than problems caused only by issues near one of our sources. In a study of four months of daily traceroutes from 30 PlanetLab vantage points to 110,000 prefixes, we found that most of the time, all but a few of the 30 traceroutes reached the prefix's origin AS. If less than 90% reached, though, it was likely the problems were more widespread; in half those cases, at least half of the traceroutes failed to reach. We use this value as our conservative threshold: if at least 90% of probes reach the origin AS of a prefix, then we assume that any probes that did not reach may have experienced congestion or problems near the source, and we ignore the issue.

For this paper, we concern ourselves primarily with reachability problems that fit these criteria. We use 90% reachability to define if a prefix is experiencing a *reachability problem*. We define a *reachability event* as the period starting when a prefix begins to experience a reachability problem and concluding when its reachability increases to 90% or higher. Such problems are often referred to as *black holes*, but we have found the term used in varying ways; instead, we use the term *reachability event* to refer to a network anomaly manifesting as a period in which a prefix experiences reachability problems.

3 Hubble Design and Architecture

Hubble attempts to discover and track reachability problems, as well as classify the problems in real-time as they occur. We base the classification on topological characteristics meant to aid diagnosis, e.g., is all of the destination traffic through a given AS affected, or only through a particular router? Is the failure related to a path change, or is it on a path that previously worked?

3.1 Goals

We seek to build a system that can provide information about ongoing reachability problems in the Internet. We hope that our system will be helpful for operators in identifying and diagnosing problems, so our system should aid in localizing the problem. It could also be used as a critical building block for overlay detouring services seeking to provide uninterrupted service between arbitrary end-hosts in the Internet. Given these potential applications, we require the system design to be driven by the following requirements.

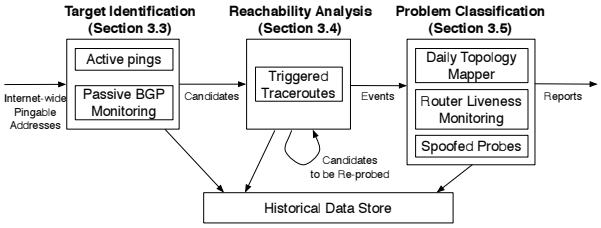


Figure 1: *Diagram of the* Hubble *architecture.*

Real-time and Continuous Information: The system should provide up-to-date and continuous information, so network operators and distributed services can quickly react to ongoing problems.

Data-plane focused: We desire a system that detects data reachability problems, regardless of whether or not they appear as BGP events. The Internet is intended to deliver data, and a BGP path on top of a broken data plane is useless. The only way to absolutely discern whether the data plane is functioning and packets can reach a destination is to send traffic, such as a traceroute, towards that destination.

Global-scale: The modern Internet is globally pervasive, and we desire a system that can monitor reachability problems for destinations of the entire Internet simultaneously, identifying most or all long-lasting reachability problems that occur. The probing and computation requirements must feasibly scale.

Light measurement traffic overhead: We intend our system to monitor areas of the Internet experiencing problems, and, in doing so, we do not want to exacerbate the problems. Our system relies on routers configured to respond to measurement probes. For these reasons, we desire a system that introduces as little measurement traffic as possible in its monitoring.

3.2 Overview of Measurement Components

As depicted in Figure 1, *Hubble* combines multiple types of measurements into four main components to identify and classify problems: pingable address discovery to decide what to monitor (not shown in figure); active ping monitoring and passive BGP monitoring to identify potential problem prefixes as targets for reachability assessment; triggered traceroutes to assess reachability and monitor reachability problems; and daily topology mapping, router liveness monitoring, and spoofed probes, combined with the same triggered traceroutes, to classify problems. The active measurements are performed using the PlanetLab infrastructure unless otherwise noted. We now present an overview of each of these measurements, and then elaborate on how the system uses these measurements to monitor and classify reachability problems.

Pingable address discovery: Pingable address discovery supplies the set of destinations for monitoring to the

active ping monitoring system. It discovers these destinations by probing the .1 in every /24 prefix present in a BGP snapshot obtained from RouteViews [31] and retaining those that respond.

Active ping monitors (details in §3.3.1): *Hubble* issues ping probes from vantage points around the world to the pingable addresses. The system in aggregate probes each address every two minutes. When a vantage point discovers a previously responsive address failing to respond, it reports the prefix as a candidate potentially experiencing more widespread reachability problems, resulting in triggered traceroutes to the prefix.

Passive BGP monitor (§3.3.2): The system observes BGP updates from multiple locations in quarter-hourly batches to maintain current AS-level routing information. This approach allows continuous monitoring in near real-time of the entire Internet from diverse viewpoints, while providing scalability by gathering information without issuing active probes. Supplementing its ping monitoring, *Hubble* analyzes the BGP updates and identifies as targets for triggered traceroutes those prefixes undergoing BGP changes, as they may experience related reachability problems. BGP feeds also allow *Hubble* to monitor prefixes in which no pingable addresses were found and hence are not monitored by the active ping monitors.

Triggered traceroute probes (§3.4.1): Every 15 minutes, *Hubble* issues active traceroute probes from distributed vantage points to targets selected as potentially undergoing problems. The system selects three classes of targets: (1) previously reachable addresses that become unreachable, as identified by the active ping monitors; (2) prefixes undergoing recent BGP changes, as identified by the passive BGP monitor; and (3) prefixes found to be experiencing ongoing reachability problems in the previous set of triggered probes.

Daily topology mapping (§3.5.1): If *Hubble* only launched traceroutes when it suspected a problem, these triggered probes would not generally give the system a view of what routes looked like before problems started. To supplement the triggered traceroutes, the system also maps the structure of the entire Internet topology using daily traceroutes, supplemented with probes to identify which network interfaces are collocated at the same router. This provides a set of baseline routes and a structured topology to map network interfaces to routers and ASes.

Router liveness monitors: Each vantage point monitors the routers on its paths from the previous day by issuing quarter-hourly pings to them. When a prefix becomes unreachable, *Hubble* uses these pings to discern whether the routers on the old path are still reachable, helping to classify what happened.

Spoofed probes (§3.5.4): Internet routes are often asymmetric, differing in the forward and reverse direction [23]. A failed traceroute signals that at least one direction is not functioning, but leaves it difficult or impossible to infer which. We employ spoofed probes, in which one monitor sets the source of packets to the IP of another monitor while probing a problem prefix. This technique aids in classification by isolating many problems to either the forward or reverse path.

3.3 Identifying Targets for Analysis

Selective targeting allows the system to monitor the entire Internet with limited active probing by identifying as targets for analysis only prefixes suspected to be experiencing problems. *Hubble* uses a hybrid approach, combining active ping monitoring with passive BGP monitoring. If *Hubble* used only passive BGP monitoring, it would miss any reachability event that did not correlate with BGP updates; as we present later in Section 4, BGP is not a good predictor of most problems, but allows *Hubble* to identify more problems than ping monitoring alone. We now present more details on how the two monitoring subsystems work.

3.3.1 Active Ping Monitoring

To meet our goal of a system with global scale, *Hubble* employs active monitoring of the reachability of prefixes. *Hubble* uses traceroute probes to perform its classification of reachability problems. However, it is not feasible to constantly traceroute every prefix in order to detect all problems. On heavily loaded PlanetLab machines, it would take any given vantage point hours to issue a single traceroute to every prefix, and so problems that were only visible from a few vantage points might not be detected in a timely manner or at all.

Hubble's active ping monitoring subsystem achieves the coverage and data-plane focus of active probing, while substantially reducing the measurement overhead versus a heavy-weight approach using pervasive traceroutes. If a monitor finds that a destination has become unresponsive, it reports the destination as a target for triggered traceroutes.

We design the ping monitors to discover as many reachability problems as possible, while reducing the number of spurious traceroutes sent to prefixes that are in fact reachable or are experiencing only transient problems. When a particular vantage point finds an address to be unresponsive, it reprobes 2 minutes later. If the address does not respond 6 times in a row, the vantage point identifies it as a target for reachability analysis, triggering distributed traceroutes to the prefix. We found that delaying the reprobes for a few minutes eliminates most transient problems, and we conducted a simple measurement study that found that the chance of a response on a

7th probe after none on the first 6 is less than 0.2%. 30 traceroutes to a destination entail around 500 total probe packets, so a 0.2% chance of a response to further pings means that it requires fewer packets to trigger traceroutes immediately, justifying launching them from distributed vantage points to investigate the problem.

A central controller periodically coordinates the ping monitors, such that (including reprobes) at least one but no more than six should probe each destination within any two minute period. Once a day, the system examines performance logs, replacing monitors that frequently fall behind or report improbably many or few unresponsive destinations. *Hubble* thus regularly monitors every destination, discovering problems quickly when they occur, without having the probing be invasive.

3.3.2 Passive BGP Monitoring

Hubble uses BGP information published by RouteViews [31] to continually monitor nearly real-time BGP routing updates from more than 40 sources. *Hubble* maintains a BGP snapshot at every point in time by incorporating new updates to its current view. Furthermore, it maintains historical BGP information for use in problem detection and analysis.

Hubble uses BGP updates for a prefix as an indicator of potential reachability problems for that prefix. In some cases, reachability problems trigger updates, as the change in a prefix from being reachable to unreachable causes BGP to explore other paths through the network. In other cases where the reachability problem is due to a misconfigured router advertising an incorrect BGP path, BGP updates could precede a reachability problem. We therefore use BGP updates to generate targets for active probes. Specifically, we select those prefixes for which the BGP AS path has changed at multiple vantage points or been withdrawn.

3.4 Real-time Reachability Analysis

Given a list of targets identified by the ping and BGP monitors, *Hubble* triggers traceroutes and integrates information from up-to-date BGP tables to assess the reachability of the target prefixes.

3.4.1 Triggered traceroutes

The daily traceroutes are of limited utility in identifying reachability problems that last only a few hours. The alternative of constantly performing traceroutes to every prefix is both inefficient and impractical. Nor do we want to sacrifice the level of detail exposed by traceroutes regarding actual routing behavior in the Internet, especially since such detail can then be used to localize the problem. *Hubble* strikes a balance by using triggered traceroutes to target prefixes identified by either the passive BGP monitor or the active ping monitors, plus prefixes known to be experiencing ongoing reachability problems. So, as long as a routing problem visible from our PlanetLab vantage points persists, *Hubble* will continually reprobe the destination prefix to monitor its reachability status.

Every 15 minutes, *Hubble* triggers traceroutes to the destinations on the target list from 30 PlanetLab nodes distributed across the globe. We limit these measurements to only a subset of PlanetLab nodes. Traceroutes from over 200 PlanetLab hosts within a short time span might be interpreted by the target end-hosts as denial of service (DoS) attacks. In the future, we plan to investigate supplementing the PlanetLab traceroutes with measurements from public traceroute servers; for example, when AS X suddenly appears in AS paths announced for a given prefix, *Hubble* could issue traceroutes to that prefix from traceroute servers that X makes available.

3.4.2 Analyzing Traceroutes to Identify Problems

In this section, we describe how *Hubble* identifies that a prefix is experiencing reachability problems. The analysis uses the triggered traceroutes, combined with *Hubble*'s passive routing view as obtained from RouteViews.

Since *Hubble* chooses as probe targets a single .1 in each of the suspect prefixes, we cannot be assured of a traceroute reaching all the way to the end-host, even in the absence of reachability problems. In some cases, a host with the chosen .1 address may not even exist, may be offline, or may stop responding to ICMP probes. Hence, we take a conservative stance on when to flag a prefix as being unreachable. We consider the origin AS for this prefix at the time when we issue the traceroute and flag the traceroute as having reachability problems if it does not terminate in the origin AS. We do this by mapping the last hop seen in the traceroute to its prefix and then to the AS originating that prefix in the BGP snapshot. Rarely, a prefix may have multiple origins [36], in which case we consider the set of ASes. We also consider the origin ASes of aliases of the last hops; if the last network interface seen on the traceroute has an alias (i.e., another IP address that belongs to the same router), and if the alias is within the address space of the origin AS, then we consider the destination reachable.

Note that, because we define reachability based on the origin AS for a prefix in routing tables, *Hubble* ignores prefixes that are completely withdrawn; these prefixes are easily classified as unreachable just by observing BGP messages. Further, note that our reachability check matches against any of the BGP paths for the prefix, rather than the particular path visible from the source of the traceroute. This is because *Hubble* issues traceroutes from PlanetLab nodes, while it gets its BGP data from RouteViews' vantage points, and the two sets are disjoint.

Traceroutes may occasionally not reach the destina-

tion for reasons that have little to do with the overall reachability of the target prefix, such as short-lived congestion on a single path or problems near the source. As described in Section 2.2, we flag a prefix as experiencing reachability problems worth monitoring only if less than 90% of the triggered probes to it reach the origin AS.

3.5 Topological Classification of Problems

As described thus far, *Hubble* only identifies prefixes experiencing problems, without pointing to the locations of the problems. To address a problem, an operator would like to know (a) the AS in which the problem occurs, to know who to contact; (b) which of the AS's routers could be causing problems; and (c) whether the issue is with paths to or from the problem prefix. Sections 3.5.2 and 3.5.3 describe how *Hubble*'s infrastructure enables classification of problems in a way that begins to address (a) and (b). In Section 3.5.4, we describe how we use spoofed probes to deal with (c). First, we describe some of the measurements that aid in classification.

3.5.1 Daily Topology Mapper

To aid in its classification, *Hubble* performs traceroutes daily to the destinations identified by the pingable address discovery. More than 200 PlanetLab sites performed traceroutes to each of these destinations once a day for the past year, and we plan to keep these daily traceroutes running continuously for the foreseeable future. These daily traceroutes enable *Hubble* to maintain a set of fairly recent base paths from each host to each destination. These base paths provide a comparison when a problem occurs; if we probed a prefix only when it was having problems, we might not know how the working paths looked before the problem. When a problem develops, *Hubble* can ping routers along the old path to determine if they remain reachable.

Additionally, we use information from the daily traceroutes to identify router interfaces, which are IP addresses belonging to different interfaces on the same router. To collect this information, we identify a list of about 2 million interfaces from our daily traceroutes and from pings to all our pingable addresses with the record route IP option enabled. The traceroutes return incoming interfaces on routers on the paths from our vantage points to the destinations, whereas the record route option-enabled pings return outgoing interfaces on routers. We identify alias candidates among these using the Mercator technique [11] (for which we probe all the interfaces using UDP probes) and the heuristic that interfaces on either end of a link are commonly in the same /30 prefix. We then probe each pair of alias candidates in succession with UDP and ICMP probes. We classify a pair of interfaces as aliases only if they return similar IP-IDs and similar TTLs [26]. *Hubble* uses this alias information in analyzing prefix reachability, as discussed in Section 3.4.2.

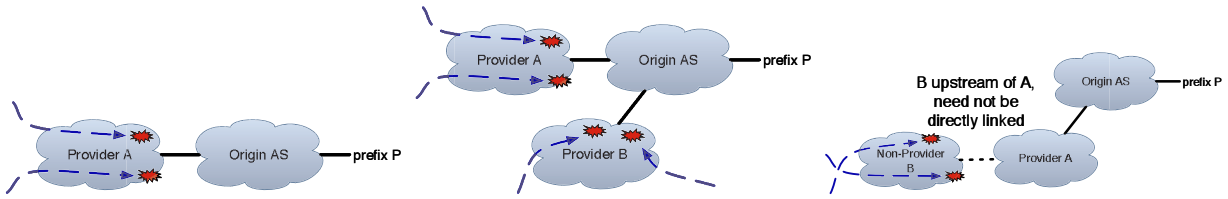
3.5.2 Hubble's Approach to Classification

Without access to complete topologies, direct BGP feeds from every AS, real-time status of router queues, and router configurations, it is often impossible to pinpoint the exact reason a given probe fails to reach its destination. Our approach with *Hubble* is to identify network entities (ASes, routers, links, or interfaces) that seem to explain the failure of a substantial number of probes to a given prefix in a round of probes. We define a reachability problem as when traceroutes from less than 90% of vantage points reach the origin AS for the prefix, so we say that an entity explains a substantial number of failed probes if it accounts for 10% or more of the vantage points. We do not require it to explain all failed probes in the set, and we may classify a problem prefix in multiple ways at once. Multiple classifications could indicate multiple simultaneous problems, multiple problems with a single root cause, or evolving problems as operators or automatic processes react or problems cascade.

Hubble's simple classification scheme relies on grouping failed probes based on the last observable hop, the expected next hop, and the ASes of each of these hops. *Hubble* infers the next hop from its historical record of working paths. We emphasize that the approach does not necessarily pinpoint the exact entity responsible; the problem could, for instance, occur on the handoff from the entity or on the return path. We address this second issue in Section 3.5.4. Our classification goals are modest– we do not currently assign blame, but simply illustrate where the traceroutes terminate. We believe this information provides a useful starting point for operators determining the exact cause of problems. In the future, we plan to expand *Hubble*'s classification abilities, as well as provide verification based on known outages and communications with operators.

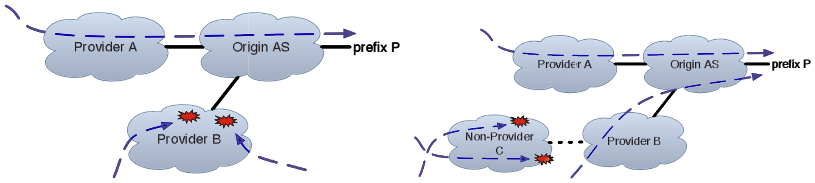
3.5.3 Classes

Hubble currently automatically assigns, in real-time, reachability problems into 9 classes, when appropriate. These classes represent different topological patterns of which traceroutes reach and which fail to reach, and they were based on preliminary hand-analysis of observed problems and chosen because they appeared to cover a substantial number of cases. Note that we infer origin and provider ASes based on active routes for the prefix in our BGP tables during the time period of the probes. In the following discussion of the classes, the destination is in prefix *P*, originated by AS *O*. We say that a probe reaches an AS if the longest matching prefix of an interface observed in the traceroute is originated by the AS,



2(a) Single-homed Origin AS Down 2(b) Multi-homed Origin AS Down 2(c) Provider(s) Unreachable

Figure 2: *Classes of complete unreachability, meaning all traceroutes fail to reach the origin AS. In (a) the origin AS has a single announced provider for the prefix, whereas in (b) it has at least 2. In both cases, some traceroutes have a hop within the provider(s). In (c) all traceroutes terminate before the provider(s).*



3(a) Provider AS Problem for Multi-homed 3(b) Non-Provider AS Problem

Figure 3: *Classes of partial reachability in which all traceroutes reaching a particular AS fail, whereas some paths work through other ASes. In (a) the AS is a provider for the origin AS, whereas in (b) it is not.*

or if one of the interfaces observed in the traceroute is an alias for an address originated by the AS.

The first three classes are cases of complete unreachability, with no traceroutes reaching even the origin AS for the prefix. They are illustrated in Figure 2.

Single-homed Origin AS Down: In this classification, none of the probes reach O, but some of the probes reach O's provider, A. Further, active AS paths for the prefix contain A as the only upstream provider for O.

Multi-homed Origin AS Down: This classification is the same as the previous, except that O has more than one provider in active BGP paths for P.

Provider(s) Unreachable: In this classification, none of the traceroutes reach the provider(s) of O, and a substantial number terminate in an AS further upstream.

Whereas in the previous classes no probes reach the prefix, the next five cover cases when some do. In the next two, all traceroutes reaching a particular AS terminate there. They are illustrated in Figure 3(a) and (b).

Provider AS Problem for Multi-homed: In this classification, all probes that reach a particular provider B of origin AS O fail to reach O, but some reach P through a different provider A. This classification is particularly interesting because ASes generally multi-home to gain resilience against failure, and an occurrence of this class may indicate a problem with multi-homed failover.

Non-Provider AS Problem: In this classification, all probes that reach some AS C fail, where C is not a direct provider of O but rather is somewhere further upstream. Some probes that do not traverse C successfully reach P.

The previous five classes represent cases in which all probes that reach some AS fail to reach the prefix. In the next two classes, all probes that reach a particular router R fail to reach P and have R as their last hop, but some probes through R's AS successfully reach P along other paths. These classes are illustrated in Figure 4(a) and (b).

Router Problem on Known Path: In this classification, R appeared on the last successful traceroute to P from some vantage point, and so the historical traceroute suggests what the next hop should be after R.

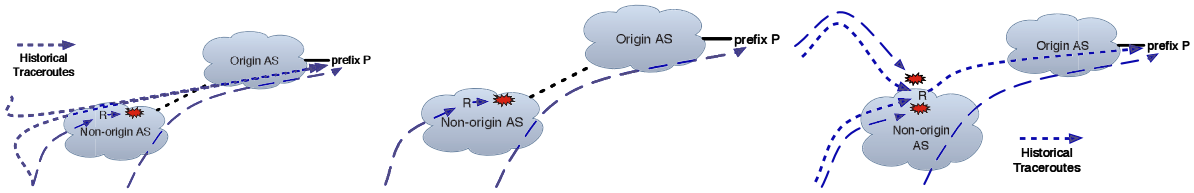
Router Problem on New Path: This classification is similar to the last, the difference being that R did not appear on the last successful traceroute to P from any vantage point. So, the problem may be due to a path change or a failure on the old path.

Next Hop Problem on Known Paths: In this classification, illustrated in Figure 4(c), no last hop router or AS explains a substantial number of failed probes. However, based on the last successful paths from some vantage points, the probes that should have converged on a particular next hop all terminated right before it.

We defined the previous 5 classes for cases of partial reachability in which some of the probes reach the prefix. All 5 have analogous versions in which some probes reach the origin AS, but none reach the prefix. We consider these less interesting, as the prefix is either down or a problem exists within the origin. So we classify them together as **Prefix Unreachable**.

3.5.4 Differentiating Failures using Spoofed Probes

The classes described above help guide an operator in searching for the causes of outages, but still leave open various explanations. Consider Figure 3 (a), with a probe



4(a) Router Problem on Known Path 4(b) Router Problem on New Path 4(c) Next Hop Problem on Known Paths

Figure 4: *Classes of partial reachability in which some paths through an AS work and others do not. Lines with shorter dashes indicate the last successful traceroute from some vantage point, whereas longer dashes indicate traceroutes from the current round of probes. In (a) and (b), all traceroutes reaching a particular router fail. In (a) the router is on known paths, in (b) on a new path. In (c) paths that previously converged at a router and reached the prefix now stop just before that router.*

from monitor a reaching through provider A, and probes from monitors b_1 and b_2 terminating with last hops in B. One might assume that the problem is between B and the origin, but it could also be a problem on the return paths to b_1 and b_2. With just the forward path information supplied by traceroutes, these cases are indistinguishable. We employ spoofed probes to differentiate the cases and provide much more specific information about the failure. Note that we only ever spoof packets using the source address of one of our other vantage points.

To determine why b_1 cannot reach P, monitor a sends probes to P with the source set as b_1. These probes reach P along a's forward path. If the responses to these probes reach b_1, then we know that the reverse path from P to b_1 works, and we determine that the failure is on b_1's forward path. Otherwise, b_1 sends probes to P with the source set as a. If b_1's forward path works, then the responses to these probes should reach a, and we determine that the failure is on the reverse path back to b_1. A central controller coordinates the spoofing, assigning, for instance, a to probe P spoofing as b_1, then fetching the results from b_1 for analysis. Because we only draw conclusions from the receipt of spoofed probes, not their absence, we do not draw false conclusions if b_1 does not receive the probe for other reasons or if the controller is unable to fetch results.

Currently, the PlanetLab kernel does not allow spoofed probes to be sent, and so spoofing is not fully integrated into *Hubble*. In Section 5.3, we provide preliminary results using a parallel deployment on RON [1], plus discuss possibilities for better future integration.

3.6 Problem Granularity and Probe Rate

In an early version of *Hubble*, with target selection based only on RouteViews updates, which are published in quarter-hourly batches, we discovered a surprising number of long-lasting problems and decided that this coarseness was still interesting. We then based our design on techniques that suffice for this granularity, with probing in quarter-hourly rounds. Besides providing abundant problems, we have other reasons for favoring long-lasting problems. First, short problems have been studied and are expected during BGP convergence [18]. Second, short-lasting problems, by definition, already resolve quickly, so are less in need of a system to identify them to operators.

It is worth considering what it would take for a future version of *Hubble* to discover shorter problems. With PlanetLab having a few hundred sites, we have a greater than 80% chance of discovering any reachability problem after 15 sites have probed the prefix, as by our definition it must be unreachable from $> 10\%$ of sites. (Since probing order is random, we consider probes to be independent.) To reliably discover minute-long problems, then, would require probing each destination every 4 seconds, a rate sustainable with our current deployment. An order of magnitude faster than that would likely require retooling the system, given the limited number of PlanetLab sites and the high load on them. However, archives of the PlanetLab Support mailing list reveal that experiments pinging destinations 4 times per second generate multiple complaints. It is unclear to us what probe rates are tolerable to ISPs.

Hubble could maintain the level of traceroutes necessary for minute-long problem discovery by increasing the number of parallel probing threads on each site and the number of sites used for triggered traceroutes (such that different sets of vantage points probe different problems). Further, a non-dedicated machine with a single Intel 3.06 GHz Xeon processor and 3 GB RAM can execute a round of reachability analysis (as MySQL scripts) in about a minute. We suspect we could significantly reduce it to keep pace with a faster probe rate, as we have not sought to optimize it given that this running time suffices for our current needs. Further, although cross-prefix correlation is an interesting future direction, current analysis is per-prefix, so the load could easily be split across multiple database servers.

4 Evaluation

Hubble is now a continuously running, automated system that we plan to keep up, minus maintenance and upgrades. We start by giving an example of one of the problems *Hubble* found. On October 8, 2007, at

5:09 a.m. PST, one of *Hubble*'s ping monitors found that *128.9.112.1* was no longer responsive. At 5:13, *Hubble* triggered traceroutes from around the world to that destination, part of *128.9.0.0/16*, originated by USC (AS4). 4 vantage points were unable to reach the origin AS, whereas the others reached the destination. All of the failed probes stopped at one of two routers in Cox Communications (AS22773), one of USC's providers, whereas the successful probes traversed other providers. In parallel, 6 of 13 RON vantage points were unable to reach the destination, with traceroutes ending in Cox, while the other 7 RON nodes successfully pinged the destination. *Hubble* launched pings from some of those 7 nodes, spoofed to appear to be coming from the other 6, and all 6 nodes received responses from *128.9.112.1*. This result revealed that the problems were all on forward paths to the destination, and *Hubble* determined that Cox was not successfully forwarding packets to the destination. It continued to track the problem until all probes launched at 7:13 successfully reached the destination, resolving the problem after 2 hours. A snapshot of the problems *Hubble* is currently monitoring can be found at `http://hubble.cs.washington.edu`.

In this section, we evaluate many of our design decisions to assess *Hubble*'s efficacy. In Section 5 we present results of a measurement study conducted using it.

How much of the Internet does *Hubble* monitor? *Hubble* selects targets from BGP updates for the entire routing table available from RouteViews. Its active ping monitoring includes more than 110,000 prefixes discovered to have pingable addresses, distributed over 92% of the edge ASes, i.e., ASes that do not provide routing transit in any AS paths seen in RouteViews BGP tables. These target prefixes include 85% of the edge prefixes in the Internet and account for 89% of the edge prefix address space, where we classify a prefix as non-edge if an address from it appears in any of our traceroutes to another prefix. Previous systems that used active probes to assess reachability managed to monitor only half as many ASes over 3 months and only when clients from those ASes accessed the system [35], whereas *Hubble* probes each of its target prefixes every 2 minutes.

We next gauge whether *Hubble* is likely to discover problems Internet users confront. To do so, we collected a sample of BitTorrent users by crawling popular sites that aggregate BitTorrent metadata and selecting 18,370 target swarms. For a month starting December 20, 2007, we repeatedly requested membership information from the swarms. We observed 14,380,622 distinct IPs, representing more than 200 of the nearly 250 DNS country codes. We are interested in whether the routing infrastructure provides connectivity, and so *Hubble* monitors routers rather than end-hosts, which are more likely to go offline (and do not affect others' reachability when they do) and often are in prefixes that do not respond to pings. Further, a router generally uses the same path to all prefixes originated by a given AS [19]. Therefore, we assess whether these representative end-users gathered from BitTorrent share origin ASes with routers monitored by *Hubble*. We find that 99% of them belong to ASes containing prefixes monitored by *Hubble*.

How effective are *Hubble*'s target selection strategies? To reduce measurement traffic overhead while still finding the events that occur, *Hubble* uses passive BGP monitoring and active ping monitoring to select targets likely to be experiencing reachability problems. Reachability analysis like *Hubble*'s relies on router-level data from traceroutes (see Sections 3.4 and 3.5). So we compare the ability of *Hubble*'s selective targeting to discover problems with an approach using pervasive tracerouting, in which the 30 vantage points each probe all monitored prefixes every 15 minutes without any target selection. We measured the total probe traffic sent by *Hubble*, including pings and traceroutes, and found that it is 5.5% of that required by the pervasive technique.

Given its much reduced measurement load, we next assess how effectively *Hubble*'s target selection strategies discover events compared to pervasive traceroutes. For this evaluation, we issued traceroutes every 15 minutes for ten days beginning August 25, 2007, from 30 PlanetLab vantage points to 1500 prefixes, and we compare the reachability problems discovered in these traceroutes with those discovered to the same set of prefixes by *Hubble*'s BGP- and ping-based target selection. We use the quarter-hourly traceroutes as "ground truth" reachability information. We only consider events that both begin and end within the experiment and only consider events that persist for at least one additional round of probing after they start. There were 1100 such reachability events, covering 333 of the prefixes, with the longest lasting almost 4 days. 236 of the events involved complete unreachability, and 874 were partial. Here and in later sections, we classify a reachability event as being complete if, at any point during the event, none of the traceroute vantage points is able to reach it. Otherwise, the event is partial.

Figure 5 shows the fraction of the events also uncovered by *Hubble*'s target selection strategies, both individually and combined. Individually, active ping monitoring uncovered 881 of the problems (79%), and passive BGP monitoring uncovered 420 (38%); combined, they discovered 939 (85%). For events lasting over an hour, the combined coverage increases to 95%. The average length of an event discovered by ping monitoring is 2.9 hours, whereas the average length of an event discovered by BGP monitoring and not by ping monitoring is only 0.8 hours.

This experiment yields a number of interesting con-

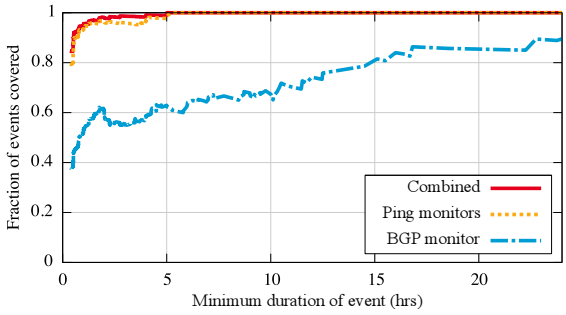


Figure 5: *For reachability events discovered in 10 days of quarter-hourly probes, fraction of events also discovered by* Hubble*'s target selection. While BGP alone proved ineffectual,* Hubble*'s techniques combine to be nearly as effective as a heavy-weight approach with the same time granularity.*

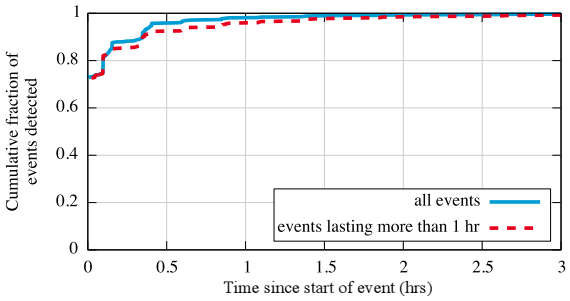


Figure 6: *For reachability events in 10 days of quarter-hourly probes, time from start of event until* Hubble *identifies its prefix as a target. Events over an hour are also given separately. Fractions are out of only the events eventually identified,* 85% *overall and* 95% *of those longer than an hour.* Hubble *identifies* 73% *of events immediately.*

clusions. First, BGP monitoring is not sufficient. We were surprised at how low BGP-based coverage was; in fact, we had originally intended to only do BGP based monitoring, until we discovered that it uncovered too few events. Second, BGP monitoring provides an important supplement to active monitoring, particularly with short events. Because we strive to limit the rate at which we probe destinations, an inherent tradeoff exists between the number of monitors (more yielding a broader viewpoint) and the rate at which a single monitor can progress through the list of ping targets. In our current implementation, we use approximately 100 monitor sites, and it takes a monitor over 3 hours to progress through the list. Therefore, short reachability problems visible from only a few vantages may not be discovered by ping monitors. BGP monitoring often helps in these cases. Third, *Hubble*'s overall coverage is excellent, meaning it discovers almost all of the problems that a pervasive probing technique would discover, while issuing many fewer probes.

How quickly after they start does *Hubble* identify problems? Besides uncovering a high percentage of all reachability events, we desire *Hubble* to identify the events in a timely fashion, and we find that it does very well at this. For the same reachability events as in Figure 5, Figure 6 shows the delay between when the event starts in the quarter-hourly probes and when the prefix is identified as a target by *Hubble*'s target selection. Because of the regular nature of the quarter-hourly probes, we know the actual starting time of the event to within that granularity. However, it is possible that *Hubble*'s monitoring identifies problems before the "continuous" traceroutes; in these cases, for ease of readability, we give the delay as 0. We additionally plot events lasting longer than an hour separately to avoid the concern that the large number of events shorter than that might distort *Hubble*'s performance. The ideal plot in the graph would be a vertical line at 0; *Hubble* achieves that for 73% of the events it identifies, discovering them at least as early as quarter-hourly probes. Of the events lasting over an hour, *Hubble* discovers 96% of them within an hour of the event's start. So *Hubble*'s light-weight probing approach still allows it to discover events in a timely fashion, and we can generally trust the duration it gives for the length of an event.

Does *Hubble* discover those events that affect sites where it does not have a vantage point? One limitation of the evaluation so far is that the 30 PlanetLab sites used to issue the quarter-hourly traceroutes are also used as part of the ping monitoring. We would like *Hubble* to identify most of the reachability problems that any vantage points would experience, not just those experienced by its chosen vantages. To partially gauge its ability to do this, we assess the quality of its coverage when we exclude the traceroute vantage points from the set of ping monitors. This exclusion leaves *Hubble* with only about $\frac{2}{3}$ of its normal number of monitors, and the excluded vantage points include 4 countries not represented in the remaining monitors. Yet our system still discovers 77% of the 1110 reachability events (as compared to 85% with all monitors). If we instead exclude an equal number of vantage points chosen randomly from those not issuing traceroutes, we see 80% coverage (median over 3 trials). We acknowledge that known diversity issues with PlanetLab somewhat limit this experiment.

To assess this limitation, we evaluate the diversity of paths seen from PlanetLab compared to BGP paths from the RIPE Routing Information Service [30], which is similar to RouteViews but with many more AS peers (447 total). The research community believes Internet routes are generally valley-free, with "uphill" and "downhill" segments. For each RIPE path, we consider only the segment from the highest degree AS to the prefix (the downhill portion); we truncate in this manner because we found more than 90% of failed traceroutes terminate within two AS hops of the destination's origin

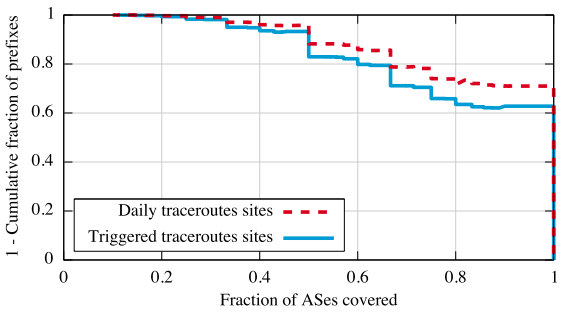


Figure 7: *Fraction of ASes on BGP paths from RIPE RIS route collectors that also appear on traceroutes from* Hubble*'s daily and triggered traceroute vantage points. Using only 35 sites for triggered traceroutes,* Hubble *observes most of the ASes visible from the 218 PlanetLab sites and the 447 RIPE peers.*

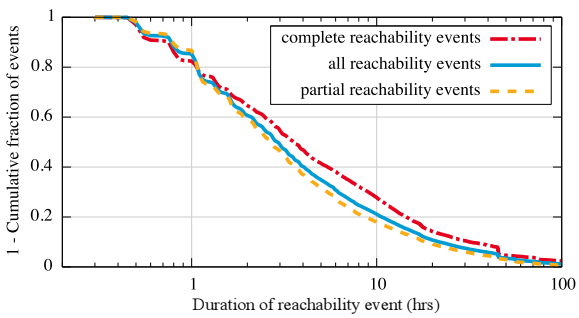


Figure 8: *CCDF of duration of reachability events.*

AS, and the relatively small number of PlanetLab sites limits the source-AS diversity on the uphill portion of paths. For each prefix monitored by *Hubble*, we consider the set of all ASes that appear on the truncated RIPE paths for that prefix. We also calculate the set of ASes that appear on one day's worth of *Hubble*'s daily traceroutes to each prefix. Figure 7 shows the fraction of ASes on RIPE paths also seen in daily traceroutes. Even with just the 35 sites used for *Hubble*'s triggered traceroutes, for 90% of prefixes, the probes include at least half of the ASes seen in BGP paths. For 70% of prefixes, the traceroutes include at least 70% of ASes, and they include all ASes for more than 60% of prefixes. These results suggest that PlanetLab achieves reasonable visibility into AS paths even with a small number of vantage points, so *Hubble* likely detects many of the AS problems that occur on the downhill portions of paths to its monitored prefixes. Further, limiting triggered probe traffic during problems to a small number of vantage points does not drastically reduce the system's coverage. While the system currently selects a single set of vantage points, seeking only to maximize the number of source countries without considering AS-path redundancy, we could easily modify it to use daily traceroutes to choose triggered traceroute sites on a per-prefix basis to maximize path diversity.

We have future plans to extend *Hubble*'s view which we mention briefly in Section 5.3; further analysis of RIPE BGP paths could suggest where our coverage is most lacking. Even now, three facts allow *Hubble* to discover many of the problems experienced by sites outside of its control. First, passive BGP monitoring gives *Hubble* a view into ASes outside of its control. Second, as noted in Section 2.2, when a problem exists, it is quite likely that many vantage points experience it. Third, as we will see in Section 5.2, many problems occur near the destinations, by which point paths from many diverse vantage points are likely to have converged.

5 Characteristics of Reachability Problems on the Internet

After demonstrating the effectiveness of *Hubble* in achieving our goals, we now present the results of a measurement study using *Hubble* to detect and measure reachability problems on the Internet for 3 weeks starting September 17, 2007. *Hubble* issued traceroutes from 35 PlanetLab sites across 15 countries (though only 30 at a time) and deployed ping monitors at 104 sites across 24 countries. In Section 2.2, we defined a reachability problem to be when a prefix is reachable from less than 90% of probing sites, and a reachability event is the period starting when we first identify that a prefix is experiencing reachability problems and concluding when its reachability increases to 90% or higher. We consider only events that began and ended during the study and persisted through at least one additional round of probing after being detected.

5.1 Prevalence and Duration

Hubble identified 31,692 reachability events, involving 10,224 distinct prefixes. 21,488 were cases of partial reachability, including 6,202 prefixes. 4,785 prefixes experienced periods of complete unreachability. *Hubble* detected an additional 19,150 events that either were transient or were still ongoing at the end of the study, involving an additional 6,851 prefixes. Of the prefixes that had problems, 58% experienced only a single reachability event, but 25% experienced more than 2 and 193 experienced at least 20.

Figure 8 shows the duration of reachability events. More than 60% lasted over 2 hours. From Section 4, we know the system has excellent coverage of events this long, but may miss some shorter ones. Still, this represents over 19,000 events longer than 2 hours, and 2,940 of the events lasted at least a day. Cases of partial reachability tend to resolve faster, with a median duration of 2.75 hours, $\frac{3}{4}$ of an hour shorter than for cases of complete unreachability. Even so, in 1,675 instances a prefix experienced partial reachability for over a day. We find this to be an astounding violation of global reachability.

5.2 Topological Characteristics

We conducted a study of *Hubble*'s reachability problem classification, applied to the triggered traceroutes issued in the first week of February, 2008. If a set of 30 probes indicates a prefix is experiencing reachability problems, *Hubble* attempts in real-time to automatically match the problem to one of the classes presented above. We first present a few case studies, then give quantitative results of *Hubble*'s classification. We intend the case studies to serve as examples of problems *Hubble* detects, but do not mean them to be exhaustive. *Hubble* classified these problems automatically, but we followed up by hand to get details such as the ASes involved.

Example of complete unreachability: For a prefix originated by an AS in Zimbabwe, probes to routers along previously successful paths to the prefix showed that the link to its primary provider seemed to have disappeared, and traffic was being routed through a backup provider. However, all probes terminated in this backup provider, either due to a misconfiguration in the secondary provider or due to the origin AS being down. In subsequent rounds of probing, packets started getting through to the destination only after the link to the primary provider came up again. This type of problem cannot be detected without active measurements, as the backup exported a valid AS path.

Example of partial reachability, AS problem: *Hubble* found that all probes to a particular prefix in Hong Kong that went through *FLAG Telecom* were dropped, whereas those that used other transit ASes reached the destination AS, *Hutchinson*. Of the 30 traceroutes to this destination, 11 went through *FLAG* and failed to reach the destination. This observation strongly suggests problems with the *FLAG-Hutchinson* connection.

Example of partial reachability, router problem: We saw an example of this scenario for an AS in Vietnam. Probes from 15 of our vantage points passed through the *Level 3* network, with some of the probes being dropped in *Level 3* while others reached the destination. Comparing the failed probes with earlier ones in which all 15 probes through *Level 3* were successful, we observed that the internal route within *Level 3* had changed. In the earlier successful traceroutes, packets reached a router 4.68.120.143 in the *Level 3* network and were forwarded to another router 213.244.165.238 (also in *Level 3*), and then to the destination AS. However, in the failed probes flagged as a reachability problem, packets reached router 4.68.120.143, which then sent them to another *Level 3* router 4.68.111.198, where the traceroutes terminated. This path change could have been due to load balancing or changes in IGP weights, because all the routers on the old path, including router 213.244.165.238, still responded to *Hubble*'s pings. This implies that either router 4.68.111.198 is misconfigured, or that the routing information is not consistent throughout the AS.

Class	Total %	Min %	Max%
Single-homed Origin AS Down	17	4	37
Multi-homed Origin AS Down	9	2	30
Provider(s) Unreachable	3	1	13
Provider AS Problem for Multi-homed	6	1	17
Non-Provider AS Problem	17	1	37
Router Problem on Known Path	7	1	40
Router Problem on New Path	21	1	40
Next Hop Problem on Known Paths	14	1	39
Prefix Unreachable	22	7	79

Table 1: *Percentage of problems in each class in one week of triggered traceroutes. Total column gives the percentage belonging to that class, out of the 375,775 total classified; recall that, as explained above, a problem can be classified in multiple ways. Min (Max) column gives the percentage of problems assigned to that class, out of all problems classified during the 15 minute window, for the 15 minute window with the lowest (highest) percentage for that class.*

Quantitative classification results: *Hubble* classified 375,775 of the 457,960 sets of traceroutes (82%) that indicated reachability problems during the study. The other problems were not classifiable using *Hubble*'s technique of grouping failed probes by last AS, last hop, or inferred next hop, then flagging any such entity that explains a substantial number. In those cases, every such entity either did not explain enough probes or had other probes that reached the destination through it, perhaps because the problem resolved while we were probing or because a problem existed on the return path to some vantage points and not others.

Table 1 shows how many problems were assigned to each class. *Hubble* classified 91.95% of all cases of complete unreachability, yielding almost one-third of the classified problems; especially for small ASes originating a single prefix, these may be cases when the prefix has simply been taken offline for awhile. The cases of partial reachability are more interesting, as a working physical path exists. Suppose s_1 is unable to reach d, but s_2 can. If nothing else, a path exists in which s_1 tunnels traffic to *Hubble*'s central coordinator (running at the University of Washington), to which it must have access as it reported d as unreachable, and *Hubble* tunnels traffic to s_2, which forwards it on to d. While phys-

ical failure– say, a bulldozer mistakenly cutting fiber– can cause complete unreachability, any case of partial reachability must be caused at least in part by either policy or misconfiguration. Policy-induced unreachability and misconfigurations might also help explain why BGP proved to be a poor predictor of reachability problems, as seen in Section 4.

We make two observations for the cases of partial reachability. First, we were surprised how often all traffic to a particular provider of a multi-homed AS failed when other providers worked. This result indicates that multi-homed failover may warrant further study and suggests that ASes may want to monitor their reachability through all their providers, perhaps using *Hubble*. Second, most of the router problems were on new paths; we plan further analysis of *Hubble* data to determine how often the routers on the old path were still available.

5.3 Classification Results Using Spoofed Probes

We conducted two studies on the RON testbed [1] to evaluate how effectively *Hubble*'s spoofed probes determine if a problem is due to issues with the forward path to or with the reverse path from the destination. Our studies used 13 RON nodes, 6 of which permitted spoofing of source addresses.

In the first study, we issued pings every half hour for a day to destinations in all the prefixes known by *Hubble* to be experiencing reachability problems at that time. We then discarded destinations that were either reachable from all 13 nodes or unreachable from all, as spoofed probes provide no utility in such cases. For every partially reachable destination d and for each RON node r which failed to reach d, we chose a node r' that could both reach d and send out spoofed probes. We had r' send a probe to d with the source address set to r. If r received d's response, it indicated a working reverse path back from d to r. We concluded that a problem on the forward path from r to d caused the unreachability . Similarly, in cases when a node r was able to send spoofed probes and unable to reach d, we had r send out probes to d with the source address set to that of a node r' from which d was reachable. If r' received d's response, it demonstrated a working forward path from r to d, and hence we concluded that the problem was on the reverse path from d back to r. We issued redundant probes to account for random losses.

How often do spoofed packets isolate the failed direction? We evaluated 25,286 instances in which one RON node failed to reach a destination that another node could reach; in 53% of these cases, spoofing allowed us to determine that the failure was on the forward path, and in 9% we determined the failure to be on the reverse path. These results were limited by the fact that we could only verify a working forward path from the 6 nodes capable of spoofing. Looking only at the 11,355 failed paths from sources capable of spoofing, we found the problem to be on the forward path in 47% of cases and on the reverse path in 21%. The remaining 32% may have had failures both ways, or transient loss may have caught packets. Our 68% determination rate represents a five-fold improvement over previous techniques [35], which were able to determine forward path problems in 13% of cases but not reverse path failures. In an additional 15% of cases, their technique inferred the failure of an old forward path from observing a path change, but made no determination as to why the new path had failed.

Class	Forward	Reverse	Mix	Unknown	Total
All destinations with reachability problems					
All nodes	49%	0%	1%	50%	3605
Spoofing nodes	42%	16%	3%	39%	2172
Multi-homed dests. classified as having provider problems					
All nodes	84%	0%	0%	16%	18762
Spoofing nodes	81%	0%	0%	19%	10628

Table 2: *Out of cases in which at least 3 vantage points failed to reach the destination, the %'s in which our technique using spoofed packets determined that all problems were on the forward path, all on the reverse path, or a mix of both. Also gives the % for which our system could not make a determination.*

The success of our technique at isolating the direction of failure suggests that, once we have an integrated *Hubble* deployment capable of spoofing from all vantage points, we will be able to classify problems with much more precision, providing operators with detailed information about most problems.

When multiple sites cannot reach a destination, how often do spoofed probes show all failed paths to be in the same direction? We then evaluated the same data to determine when all the reachability issues from RON nodes to a particular destination could either be blamed entirely on forward paths to the destination or on reverse paths back from the destination. In each half hour, we considered all targets to which at least one RON node had connectivity and at least three did not. We then determined, for each target, whether forward paths were responsible for all problems; whether reverse paths were; or whether each failed path could be pinned down to one direction or the other, but it varied across sources. We then repeated the experiment, but considered only sources capable of spoofing and only destinations unreachable from at least 3 of these sources. The top half of Table 2 presents the results. We determined the failing direction for all nodes in half of the cases, with nearly all of them isolated to the forward direction (note that the 1% difference accounts for cases when some of the spoofing nodes had reverse path failures while other

nodes had forward path ones). When considering just the spoofing nodes, we were able to explain all failures in 61% of cases. In 95% of those, the problems were isolated to either reverse or forward paths only, meaning that all nodes had paths to the destination or that the destination had paths to all nodes, respectively.

What is the nature of multi-homed provider problems? We conducted the second study to further determine how well spoofing can isolate problems. We used the same setup as before for two weeks starting October 8, 2007, but this time considered in each round only destinations that *Hubble* determined were experiencing provider AS problems for a multi-homed origin (see Figure 3 (a)). We chose this class of problems because operators we spoke with about our classification study from Section 5.2 wanted us to give them further information about what was causing the multi-homed provider problems we saw. In addition to the measurements from the first spoofing study, every RON node performed a traceroute to each destination, which we used to find those that terminated in the provider identified by *Hubble* as the endpoint for a substantial number of triggered traceroutes. We considered cases in which at least 3 paths from RON nodes terminated in the provider AS and determined in which cases we could isolate all failures. The bottom half of Table 2 gives the results. We determined the direction of all failures in more than $\frac{4}{5}$ of cases, and we were surprised to discover that all the problems were on the forward path. It seems that, in hundreds of instances a day, destinations across the Internet are reachable only from certain locations because one of their providers is not forwarding traffic to them.

What are the long term prospects for isolating the direction of failures? The above studies were limited to 13 RON nodes receiving spoofed probes, with 6 of them sending the probes. We have since developed the means to receive spoofed probes from RON nodes at all PlanetLab sites, allowing us to isolate forward path failures from the sites. Furthermore, PlanetLab support has discussed allowing spoofed probes from PlanetLab sites in future versions of the kernel. We have received no complaints about our probes, spoofed or otherwise, so they do not appear to be annoying operators. A major router vendor is talking to us about ways to provider better support for measurements.

The Internet's lack of source address authentication proved very useful in isolating the direction of failures. A more secure Internet design might have allowed authenticated non-source "reply-to" addresses. Even without this and with some ISPs filtering spoofed traffic from their end-users, we expect future versions of *Hubble* to provide better isolation in two ways. First, we can replace spoofed probes with probes sent out from traceroute servers hosted at ASes behind problems, similar to [2]. Second, we plan to deploy a measurement platform in various end-user applications which we expect will give us much wider coverage than any current deployment, allowing us to issue probes to *Hubble* vantage points from end-hosts in prefixes experiencing problems.

5.4 Summary

We found the extent of reachability problems to be much greater than we originally expected, with *Hubble* identifying reachability problems in around 10% of the prefixes it was actively monitoring and some of the problems lasting over a day.

The majority of reachability problems observed by *Hubble* fit into simple topological classes. Most of these were cases of partial reachability, in which a tunneling approach could utilize *Hubble* data to increase the number of vantage points able to reach the destination. Most surprisingly, we discovered many cases in which an origin AS was unreachable through one of its providers but not others, suggesting that multi-homing does not always provide the resilience to failure that it should.

6 Related Work

Most related work can be classified into three categories: passive monitoring at a global scale, active monitoring on a limited scale, and intra-domain monitoring using proprietary or specialized information and tools.

Passive BGP Monitoring: Numerous studies have modeled and analyzed BGP behavior. For instance, Labovitz et al. [18] found that Internet routers may take tens of minutes to converge after a failure, and that end-to-end disconnectivity accompanies this delayed convergence. In fact, multi-homed failover averaged three minutes. Mahajan et al. [21] showed that router misconfigurations could be detected with BGP feeds. Caesar et al. [3] proposed techniques to analyze routing changes and infer why they happen. Feldman et al. [8] were able to correlate updates across time, across vantage points, and across prefixes; they can pinpoint the likely cause of a BGP update to one or two ASes. Wang [33] examined how the interactions between routing policies, iBGP, and BGP timers lead to degraded end-to-end performance. BGP beacons [22] benefited this work and other studies. Together, these studies developed techniques to reverse-engineer BGP behavior, visible through feeds, to identify network anomalies. However, there are limits to such passive monitoring approaches. Though it is possible to infer reachability problems by passive monitoring [17], often times the presence of a BGP path does not preclude reachability problems and performance bottlenecks. Further, BGP data is at a coarse, AS-level granularity, limiting diagnosis.

Active Probing: Other studies used active probes to discover reachability problems. Paxson was the first to

demonstrate the frequent occurrence of reachability issues [23]. Feamster et al. [6] correlated end-to-end performance problems with routing updates. These and other studies [1, 32, 5, 9] are designed for small deployments that probe only between pairs of nodes, allowing detailed analysis but limited coverage. Pervasive probing systems, such as *iPlane* [20] and DIMES [25], exist, but have been designed to predict performance rather than to detect and diagnose faults. Ours is the first study we know of using spoofed packets to determine the direction of path failures, but Govindan and Paxson used them in a similar way to estimate the impact of router processing on measurement tools [10].

Intradomain Troubleshooting: Shaikh and Greenberg [24] proposed to monitor link state announcements within an ISP to identify routing problems. Kompella et al. also developed techniques to localize faults with ISP-level monitoring [15] and used active probing within a tier-1 ISP to detect black holes [14]. Wu et al. [34] used novel data mining techniques to correlate performance problems within an ISP to routing updates. Huang et al. [13] correlated BGP data from an AS with known disruptions; many were detectable only by examining multiple BGP streams.

Our work focuses on a previously unexplored but important design point in the measurement infrastructure space: fine-grained and continuous monitoring of the entire Internet using active probes. It enables fine-grained fault localization, modeling evolution of faults at the level of routers, and comparative evaluation of various resiliency enhancing solutions [1, 12]. Similar in spirit is Teixeira and Rexford's proposal [27], where they argue for each AS to host servers, for distributed monitoring and querying of current forwarding path state. Our work provides less complete information, due to lack of network support, but is easier to deploy. Most similar to us is PlanetSeer, which passively monitors clients of the CoDeeN CDN and launches active probes when it observes anomalies [35]. The focus of their analysis is different, providing complementary results. However, by only monitoring clients, the system covers only 43% of edge ASes and misses entirely any event that prevents a client from connecting to CoDeeN. Furthermore, this represented their aggregate coverage over 3 months, and monitoring stopped if a client had not contacted CoDeeN in 15 minutes, so some ASes may only have been monitored for brief periods. *Hubble*, on the other hand, probes prefixes in 92% of edge ASes every 2 minutes.

7 Conclusion

In this paper, we presented *Hubble*, a system that performs continuous and fine-grained probing of the Internet in order to identify and classify reachability problems in real-time on a global scale. We found that monitoring of popular BGP feeds alone does not suffice to discover most problems. At the core of our approach is a hybrid monitoring scheme, combining passive BGP monitoring with active probing of the Internet's edge prefix space. We estimate that this approach allows us to discover and monitor 85% of reachability problems, while issuing only 5.5% of the measurement traffic required by a pervasive approach with the same 15-minute granularity. In a three week study conducted with *Hubble*, we identified persistent reachability problems affecting more than 10,000 distinct prefixes, with one in five of the events lasting over 10 hours. Furthermore, two-thirds were cases of partial reachability in which a working physical path demonstrably exists.

Besides identifying problems in real-time across the Internet, we provided important early steps towards classifying problems to aid operators taking corrective action. We identified several hundred prefixes that seem not to be getting the protection that multi-homing is meant to provide; they experienced partial connectivity events where routes terminated in black holes at one provider, but were successful through another. We evaluated a prototype system that uses spoofed probes to solve the difficult problem of differentiating between forward and reverse path failures. In cases to which it fully applied, it worked five times more often than previous techniques. Applying this technique to the multi-homing cases, we isolated the direction of failure for four-fifths of problems and found all to be failures on the forward path to the prefix in question. We believe that in the future we can build on this work to deliver to operators the information they need to dramatically improve global reachability, as well as apply our system to identifying and diagnosing more general performance problems.

Acknowledgments

We gratefully acknowledge Paul Barham, our shepherd, and the anonymous NSDI reviewers for their valuable feedback on earlier versions of this paper. This research was partially supported by the National Science Foundation under grants CNS-0435065, CNS-0519696, and MRI-0619836.

References

[1] D. G. Anderson, H. Balakrishnan, M. F. Kaawhoek, and R. Morris. Resilient Overlay Networks. In *SOSP*, 2001.

[2] R. Bush, J. Hiebert, O. Maennel, M. Roughan, and S. Uhlig. Testing the reachability of (new) address space. In *INM*, 2007.

[3] M. Caesar, L. Subramanian, and R. H. Katz. Root cause analysis of Internet routing dynamics. Technical report, Univ. of California, Berkeley, 2003.

[4] D. D. Clark. The design philosophy of the DARPA Internet protocols. In *SIGCOMM*, 1988.

[5] A. Dhamdhere, R. Teixeira, C. Dovrolis, and C. Diot. NetDiagnoser: Troubleshooting network unreachabilities using end-to-end probes and routing data. In *CoNEXT*, 2007.

[6] N. Feamster, D. G. Andersen, H. Balakrishnan, and M. F. Kaashoek. Measuring the effects of Internet path faults on reactive routing. In *SIGMETRICS*, 2003.

[7] N. Feamster and H. Balakrishnan. Detecting BGP configuration faults with static analysis. In *NSDI*, 2005.

[8] A. Feldmann, O. Maennel, Z. M. Mao, A. Berger, and B. Maggs. Locating Internet routing instabilities. In *SIGCOMM*, 2004.

[9] F. Georgatos, F. Gruber, D. Karrenberg, M. Santcroos, A. Susanj, H. Uijterwaal, and R. Wilhem. Providing active measurements as a regular service for ISPs. In *PAM*, 2001.

[10] R. Govindan and V. Paxson. Estimating router ICMP generation delays. In *PAM*, 2002.

[11] R. Govindan and H. Tangmunarunkit. Heuristics for Internet map discovery. In *INFOCOM*, 2000.

[12] K. P. Gummadi, H. V. Madhyastha, S. D. Gribble, H. M. Levy, and D. Wetherall. Improving the reliability of Internet paths with one-hop source routing. In *OSDI*, 2004.

[13] Y. Huang, N. Feamster, A. Lakhina, and J. J. Xu. Diagnosing network disruptions with network-wide analysis. In *SIGMETRICS*, 2007.

[14] R. Kompella, J. Yates, A. Greenberg, and A. Snoeren. Detection and localization of network black holes. In *INFOCOM*, 2007.

[15] R. R. Kompella, J. Yates, A. Greenberg, and A. C. Snoeren. IP fault localization via risk modeling. In *NSDI*, 2005.

[16] N. Kushman, S. Kandula, and D. Katabi. Can you hear me now?! It must be BGP. *CCR*, 37(2):75–84, 2007.

[17] C. Labovitz, A. Ahuja, and M. Bailey. Shining Light on Dark Address Space. `http://www.arbornetworks.com/dmdocuments/dark_address_space.pdf`.

[18] C. Labovitz, A. Ahuja, A. Bose, and F. Jahanian. Delayed Internet routing convergence. In *SIGCOMM*, 2000.

[19] A. Lambert, M. Meulle, and J.-L. Lutton. Revisiting interdomain root cause analysis from multiple vantage points. In *NANOG*, number 40, June 2007.

[20] H. V. Madhyastha, T. Isdal, M. Piatek, C. Dixon, T. Anderson, A. Krishnamurthy, and A. Venkataramani. iPlane: An information plane for distributed services. In *OSDI*, 2006.

[21] R. Mahajan, D. Wetherall, and T. Anderson. Understanding BGP misconfiguration. In *SIGCOMM*, 2002.

[22] Z. M. Mao, R. Bush, T. G. Griffin, and M. Roughan. BGP beacons. In *IMC*, 2003.

[23] V. Paxson. End-to-end routing behavior in the Internet. *IEEE/ACM Transactions on Networking*, 1997.

[24] A. Shaikh and A. Greenberg. OSPF monitoring: Architecture, design and deployment experience. In *NSDI*, 2004.

[25] Y. Shavitt and E. Shir. DIMES: Let the Internet measure itself. *CCR*, 35(5):31–41, 2005.

[26] N. Spring, M. Dontcheva, M. Rodrig, and D. Wetherall. How to resolve IP aliases,. Technical report, Univ. of Washington, 2004.

[27] R. Teixeira and J. Rexford. A measurement framework for pinpointing routing changes. In *ACM SIGCOMM workshop on Network Troubleshooting*, 2004.

[28] `http://isotf.org/mailman/listinfo/outages`. Outages mailing list.

[29] `http://www.merit.edu/mail.archives/nanog/`. North American Network Operators Group mailing list.

[30] `http://www.ripe.net/ris/`. RIPE Routing Information Service.

[31] `http://www.routeviews.org/`. RouteViews.

[32] F. Wang, N. Feamster, and L. Gao. Quantifying the effects of routing dynamics on end-to-end Internet path failures. Technical report, Univ. of Massachusetts, Amherst, 2005.

[33] 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. *CCR*, 36(4):375–386, 2006.

[34] J. Wu, Z. M. Mao, J. Rexford, and J. Wang. Finding a needle in a haystack: Pinpointing significant BGP routing changes in an IP network. In *NSDI*, 2005.

[35] M. Zhang, C. Zhang, V. S. Pai, L. Peterson, and R. Wang. PlanetSeer: Internet path failure monitoring and characterization in wide-area services. In *OSDI*, 2004.

[36] X. Zhao, D. Pei, L. Wang, D. Massey, A. Mankin, S. Wu, and L. Zhang. An analysis of BGP multiple origin AS (MOAS) conflicts, 2001.