A measurement framework for pin-pointing routing changes

Renata Teixeira, Jennifer Rexford

A Measurement Framework for Pin-Pointing Routing Changes

Renata Teixeira
Univ. Calif. San Diego
La Jolla, CA
teixeira@cs.ucsd.edu

Jennifer Rexford
AT&T Labs–Research
Florham Park, NJ
jrex@research.att.com

ABSTRACT

Changes in the end-to-end path between two hosts can lead to sudden changes in the round-trip time and available bandwidth, or even the complete loss of connectivity. Determining the *reason* for the routing change is crucial for diagnosing and fixing the problem, and for holding a particular domain accountable for the disruption. Active measurement tools like traceroute can infer the current path between two end-points, but not where and why the path changed. Analyzing BGP data from multiple vantage points seems like a promising way to infer the root cause of routing changes. In this paper, we explain the inherent limitations of using BGP data alone and argue for a distributed approach to troubleshooting routing problems. We propose a solution where each AS continuously maintains a view of routing changes in its own network, without requiring additional support from the underlying routers. Then, we describe how to query the measurement servers along the AS-level forwarding path from the source to the destination to uncover the location and the reason for the routing change.

Categories and Subject Descriptors

C.2.2 [**Network Protocols**]: Routing Protocols; C.2.3 [**Computer-Communication Networks**]: Network Operations

General Terms

Management,Measurement, Design, Reliability, Performance

Keywords

Network troubleshooting, root cause analysis, BGP, IGP

1. INTRODUCTION

The end-to-end path between two hosts may change for various reasons, such as equipment failures and configuration changes. In addition to transient disruptions during routing convergence, the new path may have a larger round-trip time, lower available bandwidth, smaller maximum transmission unit, more aggressive packet filtering policies, or a forwarding loop or blackhole that drops packets. When multiple destinations experience routing changes at the same time, the large shift in traffic may overload one or more links in an IP backbone network. Knowing *why* the routing change happened is necessary for network administrators to diagnose and fix persistent reachability problems, or to tune the configuration of the routing protocols to rebalance the traffic load. Determining *where* the routing change originated is crucial for having greater accountability for service disruptions in the Internet. Such accountability is important for compensating end users for violations of service-level agreements and for helping network administrators select good upstream providers and peers. In this paper, we propose a measurement framework for pin-pointing the causes of routing changes.

Active measurement tools such as traceroute [1] seem like the most natural way to diagnose a routing change. However, traceroute returns inconsistent results for paths that are changing during the measurement process; in addition, some routers do not send ICMP replies and many firewalls discard the probe packets. Also, identifying the Autonomous System (AS) associated with each hop in the path is surprisingly difficult [2]. The future deployment of more sophisticated router-level support for active measurement (e.g., the IP Measurement Protocol [3, 4]) may resolve some of these issues. However, active measurement provides a view of a path only at the time the probes are sent, requiring a high probe rate to track routing changes. More importantly, active measurements alone only reveal what part of a path has changed and where packet delay, loss, or reordering occur [5, 6], but not necessarily what *caused* the route to change and where the change *originated*.

An alternate approach is to exploit publicly-available passive measurements of routing changes in the Border Gateway Protocol (BGP). Each RouteViews [7] and RIPE-NCC [8] feed logs the advertisement and withdrawal messages received via an external BGP (eBGP) session with one router in a participating AS. Recent studies have proposed looking for patterns across AS paths, destinations, and time to pin-point the location and cause of routing changes [9, 10, 11]. However, a single topology or configuration change can lead to numerous patterns of updates, and multiple events could lead to the same sequence of routing messages [12]. Combining data from multiple vantage points reduces the uncertainty but the approach is still fraught with difficulty because some routing changes are not visible in BGP and others can lead to misleading BGP messages. One of the main contributions of this paper is to identify these problems and derive guidelines for diagnosing routing changes, as discussed in Section 2.

We argue that it is possible to use passive measurements for diagnosing routing problems if each AS contributes by solving its part of the puzzle. In Section 3, we present a strawman proposal where each AS constructs a view of its part of the routing system based on data readily available from today's routers—router configuration state, BGP update messages from border routers, the up/down

SIGCOMM'04 Workshops, Aug. 30+Sept. 3, 2004, Portland, Oregon, USA.

status of BGP sessions, and intradomain routing messages. The AS uses the information to determine whether a routing change was triggered by an internal or external cause. Rather than sending raw data to a central repository, an AS accepts queries from neighboring domains about past routing changes. To diagnose external routing changes, an AS may forward a query to the next AS in either the old or the new forwarding path. Our proposed scheme can be viewed as an approach to the "Why problem" articulated in [13] or to the "automatic error reporting" scenario in [4]. In particular, we show how to answer questions like "why did the forwarding path to destination d change?" The paper concludes in Section 4 with discussion of future research directions.

2. PUBLIC BGP DATA IS NOT ENOUGH

This section highlights the challenges of finding the root cause of routing changes through analysis of BGP update data alone. We discuss why some plausible assumptions do not hold under certain scenarios. In particular, we show that (i) many routing changes are not visible in the BGP data and (ii) a partial view of the BGP data may lead to inaccurate conclusions, and derive principles that guide our approach in the next section.

2.1 Routing Changes Not Visible in eBGP

ASes in the core of the Internet usually connect to multiple neighboring ASes, and two ASes may connect in multiple physical locations. Routers at the border of a network learn how to reach external prefixes by speaking external BGP (eBGP) with routers in neighboring ASes. Upon selecting an externally-learned route, the border router uses internal BGP (iBGP) to distribute the route to the other routers inside the AS. BGP is responsible for (i) determining the AS-level route to reach a destination prefix and (ii) for each router in an AS, selecting the best egress point for forwarding traffic toward that destination prefix. The internal path from the ingress point to the egress point is determined by an Interior Gateway Protocol (IGP), such as OSPF or IS-IS. In this subsection, we discuss three "myths" that relate to how routing changes inside an AS may impact the forwarding path without being visible via an eBGP monitoring session.

MYTH: The BGP updates from a single router accurately represent the AS.

The routers A and B in Figure 1 learn how to reach destination prefix d through eBGP and propagate that information via iBGP to all other routers in AS 1. A router invokes the BGP decision process [14] to select a single best route for the prefix. The first few steps of the decision process compare the BGP attributes, such as local preference and AS path length, of the candidate routes. Next, the router prefers an eBGP-learned route over any iBGP-learned routes. Still, multiple equally-good choices may remain. For example, in Figure 1, the routes from A and B look equally attractive to router D. D breaks the tie by selecting the BGP route with the *closest egress point*—the router with the smallest IGP path cost (i.e., router B with cost of 5). Such a routing decision is commonly called *hot-potato* routing.

Hot-potato routing implies that different routers in an AS may pick different BGP-level routes. For example, B picks the eBGP route through AS 3. Router A learns two equally-good eBGP routes and chooses (say) the one via AS 2 based on an arbitrary tie break, such as the router id. Based on hot-potato routing, router D selects the route through B and router C selects the route through A. As such, BGP data collected from D would only reveal the route via AS 3. Now suppose that a failure occurs on the link connecting router A to AS 2. Then, both A and C would switch to the route via AS 3, which may lead to a change in the properties of the end-to-end paths for traffic entering AS 1 at router C. However, the link failure does *not* cause a change in the BGP route at D and, as such, the change is not visible to the measurement system.

IMPLICATION 1. *The measurement system needs to capture the BGP routing changes from all of the border routers.*

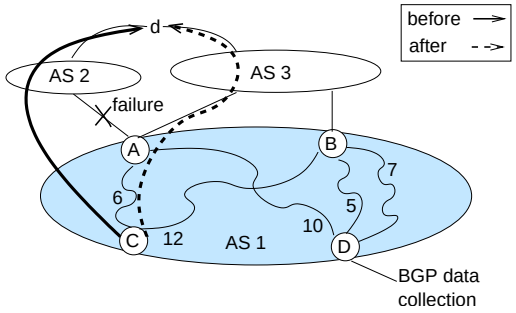

Figure 1: BGP changes are not detected at data collection point.

MYTH: Routing changes visible in eBGP have greater end-to-end impact than changes with local scope.

IGP and iBGP changes may have a significant influence on end-to-end performance without causing any eBGP-visible routing change. In Figure 1, router D has three internal paths to reach d—two via egress point B (with IGP costs of 5 and 7, respectively) and one via egress point A (with cost 10). Due to hot-potato routing, D selects the route through B with cost 5. Even if a link fails on the shortest path, D continues to use egress point B, though packet forwarding shifts to the path with cost 7. This does not cause an iBGP routing change, let alone an eBGP-visible change. Yet, if the path with cost 7 has low available bandwidth or a high round-trip time, the effects on user performance might be significant.

Suppose now that a link failure makes all paths from D to B have an IGP cost higher than 10. Then, router D switches to the BGP route with egress point A. However, this iBGP routing change may or may not be visible in eBGP. If A were routing traffic via AS 3, then D's new best BGP route would have the same AS path as the old one. Under the common practice of *non-transitive attribute filtering*, router D would *not* send a new eBGP advertisement to its neighbors. However, if A were routing traffic via AS 2, router D would need to send an eBGP update to its neighbors upon switching egress points. Either way, the traffic entering the AS at D may experience a noticeable change in performance properties.

IMPLICATION 2. *The measurement system needs to capture IGP and iBGP routing changes inside an AS.*

MYTH: BGP data from a router accurately represents routing changes on that router.

Network operators often configure their BGP-speaking routers to limit the scope of advertisements for subnets of larger address blocks, in order to limit the size of the BGP routing tables [15]. In Figure 2, router A is an access router that connects to several customer networks that have been assigned address blocks out of the larger prefix 12.1.0.0/16. For example, A may have a static route directing traffic for 12.1.1.0/24 through the access link to a specific customer. Router A does not need to advertise the 12.1.1.0/24 route to any other routers inside the AS, or to routers in other domains; instead, A simply advertises reachability to the supernet 12.1.0.0/16. Even a BGP feed collected *directly* from router A

would not reveal the existence of the $12.1.1.0/24$ subnet or any changes in the reachability of this subnet. For example, following a failure of the customer's access link, the forwarding path of traffic destined to addresses in $12.1.1.0/24$ would terminate at A. Yet, the BGP monitoring system would not observe any routing change.

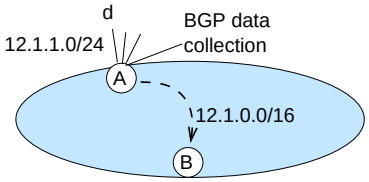

Figure 2: Subnet 12.1.1.0/24 at router A is not visible in BGP

In addition to the example in Figure 2, other prefixes may be invisible due to the BGP export policies applied on the monitoring session. For example, an AS may export customer-learned routes to a public monitoring system but not the routes learned from private peers; often the exact details of which routes an AS exports to the RouteViews and RIPE-NCC monitors are unknown.

IMPLICATION 3. *The measurement system needs to know all routes the router knows, even if they are not normally visible in eBGP.*

2.2 Misleading BGP Changes

Recent studies [9, 10, 11] propose techniques for analyzing patterns in the BGP updates from multiple vantage points to infer the location and cause of routing changes. The algorithms cluster the data by time, prefix, and AS path to discover common explanations for a set of BGP updates. The accuracy of these techniques depends on the completeness of the input data. In this subsection, we discuss how partial BGP data can lead to incorrect diagnosis of a routing change.

MYTH:The AS responsible for a BGP routing change appears in the old or the new AS path [9, 10, 11].

The inference algorithms build on the assumption that the AS responsible for a routing change appears in either the old path, the new path, or both. However, this may not hold when some of the ASes in the forwarding path do not contribute BGP feeds. In the example in Figure 3, suppose that the sideways links between these ASes are private peering links, where each AS exports only the BGP routes learned from its downstream customers [16]. All other links in the system correspond to provider-customer relationships where each AS exports its best route for each prefix. For simplicity, assume that each AS selects the BGP route with the shortest AS path, among the choices learned from the neighbors. In Figure 3(a), ASes 1, 2, and 3 all choose the path through AS 2; in particular, AS 1 prefers the path through AS 2 over the longer path via AS 4.

Now, suppose that AS 11 becomes a customer of AS 3, as shown in Figure 3(b). In response to this event, AS 3 now selects the new shorter AS path through AS 11 and announces the new path to AS 2. AS 2 prefers the new path over the old path through AS 8 and starts directing traffic via AS 3. This causes AS 2 to withdraw the BGP route it had advertised earlier to AS 1. Note that AS 2 does *not* advertise the new route to AS 1 because of the export policy (i.e., "do not export a route learned from one peer to another"). This causes AS 1 to switch to the longer customer-learned route via AS 4, as shown in Figure 3(b). Based only on BGP data from ASes 1, 4, 5, 6, and 7, the inference algorithm would only see the withdrawal of the BGP route via AS 2. From AS 1's vantage point, the AS path changes from "1 2 8 9 10" to "1 4 5 6 7 10"—ASes 3

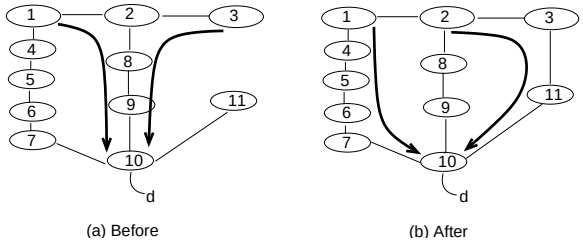

Figure 3: AS causing the routing change is not in the old or new AS paths

and 11 do not appear anywhere in either the old or new paths. Collecting measurement data from more vantage points would reduce the likelihood of these kinds of problems, but knowing how many vantage points are truly necessary is difficult without full knowledge of the AS graph and the routing policies.

IMPLICATION 4. *Accurate troubleshooting of routing changes may require measurement data from each AS.*

MYTH: Looking at routing changes across prefixes resolves ambiguity about the origins of a routing change.

The inference algorithms narrow down the origin of a routing change by identifying the common attributes for prefixes that experience a routing change close together in time. In Figure 4, suppose that each AS has a "shortest AS path" routing policy. Router C in AS 1 has two BGP-learned routes to reach destination $d2$ and initially selects egress point A because of hot-potato routing. If the cost of the IGP path from C to A increases to 11, then C would select egress point B to route to $d2$. In contrast, the BGP routes for $d1$ and $d3$ would not change because AS 1 has a single egress point for reaching each of these destinations.

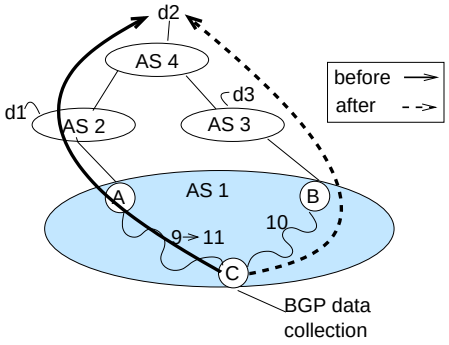

Figure 4: Internal routing change affecting only destinations in AS 4

This hot-potato routing change could be misleading to an external observer. If AS 4 originates multiple destinations, the BGP update stream from C would show many routes changing AS paths from "1 2 4" to "1 3 4". This would suggest that one of the four ASes is involved. By looking across all prefixes, the observer would see that all destinations originated by AS 4 shift at the same time, and those originated by ASes 2 and 3 do not change. This could lead to the incorrect inference that AS 4 (or the link between AS 4 and AS 2) is responsible for the change. Large hot-potato routing changes (such as reported in [17]) may also be mistakenly associated with a BGP session reset in one of the links in the AS path.

IMPLICATION 5. *The ASes involved in the routing change should cooperate to pin-point the reason for the routing change.*

MYTH: The BGP signaling path is an accurate representation of the AS-level forwarding path.

Analysis of changes in the BGP AS paths does not necessarily shed light on the changes in the forwarding path because the two paths do not necessarily match [2]. For example, route aggregation may result in a BGP AS path that does not include the AS(es) at the end of the forwarding path. In addition, the iBGP configuration inside an AS may lead to packet *deflections* where one router forwards a packet to another router that has a different AS path for the same prefix [18]. These deflections may in fact be the root cause of a routing anomaly, making it important to have an accurate view of the real forwarding path. Finally, configuration mistakes (whether accidental or intentional) can lead to an incorrect BGP AS path. For example, an operator may configure a router to perform AS *prepending* (the common practice of adding artificial hops in the BGP AS path) with the wrong AS number. This can lead to a BGP AS path that bears little resemblance to the actual AS-level forwarding path. These mismatches between the two paths can lead to faulty conclusions. For example, real changes in the forwarding path might not be visible as BGP routing changes, and vice versa. Fortunately, each AS has enough internal information to know the *next-hop* AS in the AS-level forwarding path.

IMPLICATION 6. *Troubleshooting of routing changes needs to propagate hop-by-hop along the AS-level forwarding path.*

The accuracy of identifying the root cause of routing changes using public BGP data depends on how often these myths are violated and how much coverage is need to get accurate results. Validating these hypothesis requires further research, using exactly the AS-level measurements that we propose in the next section.

3. PIN-POINTING ROUTING CHANGES

We draw on the insights learned from the previous section to sketch a distributed troubleshooting service. Implications 1 and 3 imply that we need a better source of data that represents the AS-level BGP routing decisions (an "AS-level forwarding table", if you will), and Implication 2 suggests that we also need to keep track of internal changes. In this section, we propose that each AS have an *Omni* server that constructs a comprehensive view of its part of the routing system[1]. Implications 4 and 5 imply the need for cooperation of the ASes involved in a routing change. Thus, the Omni in one AS may need to contact Omni servers in other ASes. Implication 6 suggests that the query resolution should follow the forwarding path; hence the Omni may launch a query to the next AS in the old or new forwarding path to the destination. After describing how the Omni server constructs the AS-level forwarding table and maintains the local routing state of the AS, we discuss the hop-by-hop propagation of queries. We end this section with a brief discussion of directions for future research.

3.1 AS-level Forwarding Table

We define an "AS-level forwarding table" as a mapping from prefixes to egress sets, where an egress set is the set of outgoing links that the border routers in the AS use to reach the prefix. The Omni needs to build an AS-level forwarding table to: (i) identify routing changes at the edge of the AS and (ii) determine which neighboring ASes to query about routing changes caused by external events. For example, in Figure 1, the Omni for AS 1 would compute the egress set $\{(A, AS\ 2), (B, AS\ 3)\}$ for destination d prior to the failure of the link to AS 2. After the failure, the set would change to $\{(A, AS\ 3), (B, AS\ 3)\}$. Instead of keeping all BGP update messages, the Omni only maintains a log of changes to the egress set. For example, the Omni would not need to retain information about BGP updates that change a downstream AS in the AS path or other route attributes.

To compute egress set changes, the Omni collects iBGP updates from *all border routers*[2]. Then, the Omni gathers the best routes for each border router to determine the egress set for each destination prefix. The AS-level forwarding table includes *all prefixes known at the router*, in order to avoid the kinds of problems depicted in Figure 2. This is accomplished by configuring the iBGP session to the Omni server to inject *all* routes that a router learns, including static routes (which might not normally be injected in to BGP) and subnets that would normally have limited scope.

The Omni can then do an on-line pre-processing of this more complete BGP update streams to compute the egress set for each prefix and store changes to this set with a timestamp. This dataset represents the AS-level view of external routing changes and could conceivably serve as an improved feed to public BGP repositories such as RouteViews or RIPE-NCC. Currently, RouteViews and RIPE-NCC receive an *eBGP* update stream from an *individual* router in the AS. Today, these eBGP streams exclude prefixes that are not injected into BGP. In addition, there is no differentiation between internally and externally learned routes, and no information about routing changes that are subject to non-transitive attribute filtering.

3.2 Identifying Local Routing Changes

The Omni server also needs to keep track of local routing state—the egress point selected by each router for each prefix, the forwarding path through the AS, and the routing changes caused by this AS. We define a *subpath* as the part of the forwarding path from the ingress router to the outgoing edge link connecting to the next AS. The Omni is responsible for determining whether a subpath has changed (*local effect*) and whether the AS was responsible for this change (*local cause*).

Upon detecting a performance or reachability problem, the source asks its local Omni if a routing change has occurred. In particular, the source s asks the Omni if ingress router i had any routing change to destination address d around time t. The Omni determines if the subpath for (i, d) changed and whether the cause was local or not, using the decision tree presented in Figure 5. First, the Omni searches for a change in the egress set for d close to time t. Upon detecting an egress-set change, the Omni determines that the routing change had local cause if there was either a *policy change* or an *edge change* (i.e., an eBGP session failure or a change for a subnet not normally injected in BGP) consistent with the routing change. Otherwise, the routing change has an external cause. If the egress set for d has not changed, the Omni determines whether the subpath from i to d has changed by examining both *iBGP* and *IGP* routing information for local causes.

The decision tree depends on the kinds of measurement data that are routinely collected for network management purposes:

- **Policy changes:** The Omni extracts the AS's policies from snapshots of the router's configuration state every time there is a change. In practice, changes occur infrequently.
- **BGP session status:** The status of BGP sessions can be obtained

[1] The name *Omni* is meant to capture the fact that the server is *omniscient* about the routing state in the domain.

[2] Routers do not need to forward routes learned via iBGP, since the Omni learns these routes directly. That is, the Omni should be configured as an iBGP "peer" of each border router, rather than a route-reflector client. This substantially reduces the number of BGP updates received by the Omni.

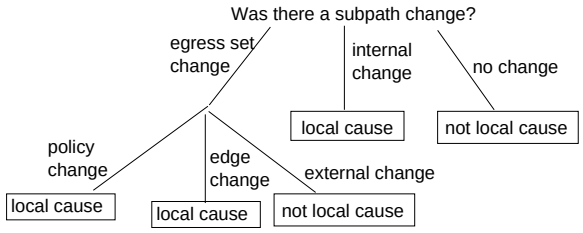

Figure 5: Omni decision tree for classifying changes in a subpath

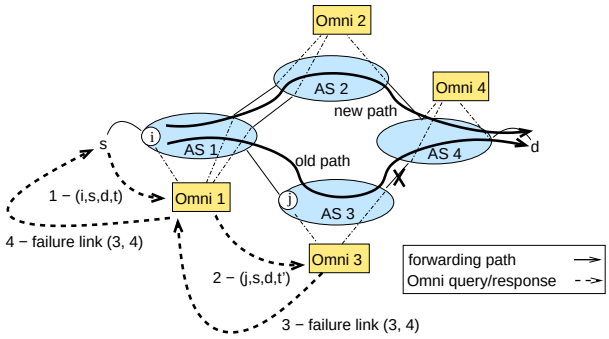

Figure 6: Collection of Omni servers diagnosing a routing change

by either SNMP data or the vendor-specific "syslog." The status of iBGP sessions is used to determine the propagation of BGP routes inside the AS, whereas the status of eBGP sessions is used to identify "edge" changes.

- **IGP changes:** An IGP routing monitor [19] can continuously track the topology (routers and links) and the IGP parameters (such as link weights). This enables the Omni to learn about changes in the forwarding paths between pairs of routers inside the AS, as well as the IGP path costs that influence the BGP-level routing decisions. The Omni ignores messages, such as refresh and duplicate IGP messages, that do not indicate a routing change.

The Omni can use the egress sets, iBGP session status, and IGP data to compute the subpath for each ingress router and destination prefix, using the model presented in [20, 21].

3.3 Inter-AS Coordination

Imagine that source s in Figure 6 is communicating with destination d when the link between ASes 3 and 4 fails. Source s asks[3] Omni 1 if the ingress router i had any routing change to destination d around time t. Following the decision tree in Figure 5, Omni 1 determines that the egress set changed because the BGP route through AS 3 was withdrawn. Recognizing that the local routing change had an external cause, Omni 1 queries Omni 3 for the reason of router j's change to destination d at the time it learned of the egress set change. Omni 3 uses its own data to determine that the failure of the eBGP session to AS 4 caused the routing change, and responds to Omni 1, which in turn responds to s.

The Omni decides how to respond to a query by identifying (i) whether the subpath changes (*local effect*) and (ii) whether the AS is responsible for the change (*local cause*):

- **Local effects and local cause:** When the AS is responsible for the routing change, the Omni responds directly to the query with an explanation.
- **Local effects and non-local cause:** When the local routing change has an external cause, the Omni examines the egress-set change to determine which neighboring ASes to query—the neighbor in the old subpath, the new subpath, or both. In the earlier example in Figure 3 in Section 2.2, the Omni in AS 1 would query the Omni AS 2 (along the old path, which has disappeared) which would, in turn, query the Omni in AS 3 which could explain the routing change.
- **No local effects:** If the Omni observes no local routing change, then the change must have an external cause. The Omni simply directs the query to the next AS in the forwarding path; since the local subpath has not changed, both the "old" and the "new" neighbor ASes are the same.

If the query reaches the AS responsible for the destination IP address, the Omni for that AS could optionally initiate a reverse query toward s to determine whether a routing change occurred on the path from d to s.

In [4], Bennett describes a scenario for automatic network error correction that resembles the behavior described here. Using IPMP, a user identifies the last working AS in the forwarding path and issues a trouble report to that AS. In this scenario, the responsibility of diagnosing the problem falls to the AS where the *effect* of the problem is observed, not the one that caused the routing change. This AS does not necessarily have enough information to diagnose the problem. In our approach, queries are propagated via Omni servers in the ASes along the forwarding path, rather than through the forwarding-plane itself. Our approach avoids the expense of placing new functionality in the forwarding plane and allows the queries to access a wider range of information about the old and new forwarding paths to pin-point the location and cause of a routing change.

3.4 Challenges for Distributed Diagnosis

Our troubleshooting scheme raises several important practical issues that warrant further discussion and investigation:

Reachability of Omni servers: We envision that each end host would know the name or IP address of the Omni servers in its own domain, and that each Omni server would know the IP addresses of the Omnis in neighboring ASes; we do not expect that this information would need to change often. For simplicity, the border routers in one AS could be configured with static routes to direct packets sent to an Omni via the edge links connecting to the neighboring AS. We envision that an AS would have multiple Omnis in different locations to reduce the likelihood that the very failure that causes a routing problem for end users compromises access to the troubleshooting service.

Scalability of Omni servers: An Omni could be overwhelmed by attack traffic or even legitimate queries. An AS can install packet filters on its edge links that discard all packets destined to the Omni that do not have a source address corresponding to an Omni in the neighboring domain. To prevent excessive queries, the edge links could impose a rate limit on traffic from each sender. In some cases, a high query rate may be indicative of a legitimate routing problem affecting multiple users. An Omni could coalesce related queries or return cached results without contacting the next AS in the path. In fact, the large number of (related) queries might provide valuable hints about the scope of a routing problem.

Time interval of a routing change: The initiator of a query can provide a time interval when a routing change may have occurred. An Omni along the query path may refine the time interval based

[3]For instance, ISPs could provide a Web interface for customers to initiate troubleshooting requests.

on its own measurement data. The measurements may reveal that multiple routing changes occur close together in time (e.g., during BGP path exploration during delayed convergence [22]). We envision that the Omni would answer queries about changes from one stable route to another, rather than reporting the short-lived routes during the transition. The Omni also needs to keep track of prefixes with routes that flap continuously to respond to queries about these destinations.

Incentives for ASes to participate: Our troubleshooting service depends on the participation of many, if not all, of the ASes in the core of the Internet. The cooperation of stub ASes would be valuable, too, to diagnose routing problems originating inside these networks. We believe ISPs would want to provide a troubleshooting service to their customers as part of a service-level agreement (SLA). These ISPs would need to have similar arrangements with their peers and upstream providers to ensure accountability for network disruptions. In fact, a collection of ASes (e.g., run by one company or consortium) could provide an SLA only for IP traffic that stays within the group of ASes, allowing for a partial deployment of Omnis. In a competitive environment, separate mechanisms are necessary to prevent ASes from providing inaccurate responses to queries. An AS could use its own BGP update data to validate the responses sent by a neighbor's Omni. More generally, third parties could use traceroute or BGP update data to detect persistently suspicious responses.

4. CONCLUSIONS

Identifying the location and cause of routing changes is crucial for troubleshooting performance and reachability problems. Currently available measurement data, such as traceroute probes and public BGP update feeds, are not sufficient. Instead, we believe that the infrastructure should have direct support for the diagnosis of routing problems. We argue that each AS should have an *Omni* server that constructs a network-wide view of its part of the Internet routing system and answers (and forwards) queries about possible routing changes. The Omni could also store information about the MTU size and packet filter for each link to diagnose other kinds of reachability problems. In addition, with traffic measurements from the edge links, the Omni server could detect shifts in incoming traffic and query the *preceding* domain about the change.

Although our solution does not rely on special support from the network, extensions to the routers such as proposed in IPMP would make the problem easier to solve. Ideally, each router would have a special monitoring session that provides a view of all of the routes it learns (including alternate BGP routes as well as routes not injected into BGP), the dynamic status of its routing protocol adjacencies (e.g., for OSPF adjacencies and BGP sessions), and an explanation for local routing changes (e.g., local policy change, withdrawal of best route by a neighbor, etc.). More generally, we believe that extending the routing protocols to reveal the underlying reason for a routing change is a promising avenue for future work.

Acknowledgments

We would like to thank Jay Borkenhagen for his invaluable insights about the challenges of diagnosing routing problems in production networks. Thanks also to Christophe Diot, Geoff Voelker, and the anonymous reviewers for their helpful comments. Renata Teixeira was supported by a fellowship from Capes/Brazil and by AT&T's support for the UCSD Center for Networked Systems.

5. REFERENCES

[1] V. Jacobson, "Traceroute." `ftp://ftp.ee.lbl.gov/traceroute.tar.gz`.

[2] Z. M. Mao, J. Rexford, J. Wang, and R. H. Katz, "Towards an accurate AS-level traceroute tool," in *Proc. ACM SIGCOMM*, August 2003.

[3] A. McGregor and M. Luckie, "IP measurement protocol (IPMP)." Internet Draft, draft-mcgregor-ipmp-04.txt, February 2004.

[4] J. Bennett, "The case for an Internet Measurement Protocol," November 11 2003. E-mail posting on the Internet Measurement Research Group, `http://www1.ietf.org/mail-archive/working-groups/imrg/current/msg00154.%html`.

[5] N. Feamster, D. Anderson, H. Balakrishnan, and F. Kaashoek, "Measuring the effects of Internet path faults on reactive routing," in *Proc. ACM SIGMETRICS*, June 2003.

[6] R. Mahajan, N. Spring, D. Wetherall, and T. Anderson, "User-level Internet path diagnosis," in *Proc. ACM SOSP*, October 2003.

[7] "Route Views." `http://www.routeviews.org`.

[8] "RIPE NCC RIS." `http://www.ripe.net/ripencc/ris`.

[9] D.-F. Chang, R. Govindan, and J. Heidemann, "The temporal and topological characteristics of BGP path changes," in *Proc. IEEE ICNP*, November 2003.

[10] M. Caesar, L. Subramanian, and R. H. Katz, "Towards localizing root causes of BGP dynamics," Tech. Rep. CSD-03-1292, UC Berkeley, November 2003.

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

[12] T. G. Griffin, "What is the sound of one route flapping?." presentation at the *Network Modeling and Simulation Summer Workshop*, 2002.

[13] D. Clark, C. Partridge, J. C. Ramming, and J. Wroclawski, "A knowledge plane for the Internet," in *Proc. ACM SIGCOMM*, August 2003.

[14] "A Border Gateway Protocol 4 (BGP-4)." Internet Draft draft-ietf-idr-bgp4-24.txt, work in progress, November 2003.

[15] E. Chen and J. Stewart, "A Framework for Inter-Domain Route Aggregation," RFC 2519, IETF, February 1999.

[16] G. Huston, "Interconnection, peering, and settlements," in *Proc. INET*, June 1999.

[17] R. Teixeira, A. Shaikh, T. Griffin, and J. Rexford, "Dynamics of hot-potato routing in IP networks," in *Proc. ACM SIGMETRICS*, June 2004.

[18] T. G. Griffin and G. Wilfong, "On the correctness of IBGP configuration," in *Proc. ACM SIGCOMM*, August 2002.

[19] A. Shaikh and A. Greenberg, "OSPF monitoring: Architecture, design, and deployment experience," in *Proc. USENIX/ACM NSDI*, March 2004.

[20] N. Feamster, J. Winick, and J. Rexford, "A model of BGP routing for network engineering," in *Proc. ACM SIGMETRICS*, June 2004.

[21] A. Feldmann, A. Greenberg, C. Lund, N. Reingold, and J. Rexford, "NetScope: Traffic engineering for IP networks," *IEEE Network Magazine*, pp. 11–19, March 2000.

[22] C. Labovitz, A. Ahuja, A. Bose, and F. Jahanian, "Delayed Internet routing convergence," *IEEE/ACM Trans. on Networking*, June 2001.