Dynamics of hot-potato routing in IP networks

Renata Teixeira, Aman Shaikh, Tim Griffin, Jennifer Rexford

Dynamics of Hot-Potato Routing in IP Networks

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

Aman Shaikh
AT&T Labs–Research
Florham Park, NJ
ashaikh@research.att.com

Tim Griffin
Intel Research
Cambridge, UK
tim.griffin@intel.com

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

ABSTRACT

Despite the architectural separation between intradomain and interdomain routing in the Internet, intradomain protocols do influence the path-selection process in the Border Gateway Protocol (BGP). When choosing between multiple equally-good BGP routes, a router selects the one with the *closest* egress point, based on the intradomain path cost. Under such *hot-potato* routing, an intradomain event can trigger BGP routing changes. To characterize the influence of hot-potato routing, we conduct controlled experiments with a commercial router. Then, we propose a technique for associating BGP routing changes with events visible in the intradomain protocol, and apply our algorithm to AT&T's backbone network. We show that (i) hot-potato routing can be a significant source of BGP updates, (ii) BGP updates can lag 60 seconds or more behind the intradomain event, (iii) the number of BGP path changes triggered by hot-potato routing has a nearly uniform distribution across destination prefixes, and (iv) the fraction of BGP messages triggered by intradomain changes varies significantly across time and router locations. We show that hot-potato routing changes lead to longer delays in forwarding-plane convergence, shifts in the flow of traffic to neighboring domains, extra externally-visible BGP update messages, and inaccuracies in Internet performance measurements.

Categories and Subject Descriptors

C.2.2 [**Network Protocols**]: Routing Protocols; C.2.6 [**Computer-Communication Networks**]: Internetworking

General Terms

Algorithms, Management, Performance, Measurement

Keywords

Hot-potato routing, BGP, OSPF, convergence

1. INTRODUCTION

End-to-end Internet performance depends on the stability and efficiency of the underlying routing protocols. A large portion of Internet traffic traverses multiple Autonomous Systems (ASes), making performance dependent on the routing behavior in multiple domains. In the large ASes at the core of the Internet, routers forward packets based on information from both the *intra*domain and *inter*domain routing protocols. These networks use the Border Gateway Protocol (BGP) [1] to exchange route advertisements with neighboring domains and propagate reachability information within the AS. The routers inside the AS use an Interior Gateway Protocol (IGP) to learn how to reach each other. In large IP networks, the two most common IGPs are OSPF [2] and IS-IS [3], which compute shortest paths based on configurable link weights. A router combines the BGP and IGP information to construct a forwarding table that maps destination prefixes to outgoing links.

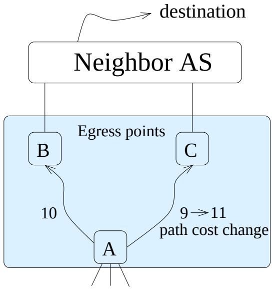


Figure 1: Hot-potato routing change from egress C to B

The two-tiered routing architecture should isolate the global Internet from routing changes within an individual AS. However, in practice, the interaction between intradomain and interdomain routing is more complicated than this. The example in Figure 1 shows an AS with two external BGP (eBGP) sessions with a neighboring AS that advertises routes to a destination prefix. The two routers B and C propagate their eBGP-learned routes via internal BGP (iBGP) sessions with router A. This leaves A with the dilemma of choosing between two BGP routes that look "equally good" (e.g., with the same number of AS hops). Under the common practice of *hot-potato* routing, A directs traffic to the closest *egress point*—the router with the smallest intradomain path cost (i.e., router C). This tends to limit the bandwidth resources consumed by the traffic by moving packets to a next-hop AS at the nearest opportunity. However, suppose the IGP cost to C changes from 9 to 11, in response to a link failure along the original path or an intentional change in a link weight for traffic engineering or planned maintenance. Although the BGP route through C is still available, the IGP cost change would cause A to select the route through egress point B. We refer to this as a *hot-potato routing change*.

Hot-potato routing changes can have a significant performance impact: (i) transient packet delay and loss while the routers recompute their forwarding tables, (ii) shifts in traffic that may cause

SIGMETRICS/Performance'04, June 12–16, 2004, New York, NY, USA.

congestion on the new paths through the network, and (iii) BGP routing changes visible to neighboring domains. The frequency and importance of these effects depend on a variety of factors. A tier-1 ISP network connects to many neighboring domains in many geographic locations. In practice, an ISP typically learns “equally good” BGP routes at each peering point with a neighboring AS, which increases the likelihood that routing decisions depend on the IGP cost to the egress points. In addition, the routers have BGP routes for more than 100,000 prefixes, and a single IGP cost change may cause many of these routes to change at the same time. If these prefixes receive a large volume of traffic, the influence on the flow of traffic within the AS and on its downstream neighbors can be quite significant. In this paper, we quantify these effects by analyzing the IGP-triggered BGP updates in AT&T’s backbone network (AS 7018).

On the surface, we should be able to study hot-potato routing changes in an analytical or simulation model based on the protocol specifications. However, the interaction between the protocols depends on details not captured in the IETF standards documents, as discussed in more detail in Section 2. Vendor implementation decisions have a significant impact on the timing of messages within each protocol. The design of the network, such as the number and location of BGP sessions, may also play an important role. In addition, the behavior of the routing protocols depends on the kinds of low-level events—failures, traffic engineering, and planned maintenance—that trigger IGP path changes, and the properties of these events are not well-understood. In light of these issues, our study takes an empirical approach of controlled, *black box* experiments at the router level coupled with a joint analysis of IGP and BGP measurements collected from a large ISP network.

Although previous studies have characterized IGP link-state advertisements [4, 5, 6, 7] or BGP update messages [7, 8, 9, 10] in isolation, we believe this is the first paper to present a joint analysis of the IGP and BGP data. The work in [9] evaluates how BGP routing changes affect the flow of traffic inside an ISP backbone but does not differentiate between routing changes caused by internal and external events. The main contributions of this paper are:

- **Evaluating hot-potato changes at the router level:** We describe how to evaluate hot-potato routing changes on a single router. We perform experiments with a Cisco router to illustrate the timing of the protocol messages in a controlled environment.
- **Identifying hot-potato BGP routing changes:** Our algorithm for correlating the IGP and BGP data (i) generates a sequence of path cost changes that may affect BGP routing decisions, (ii) classifies BGP routing changes in terms of possible IGP causes, and (iii) matches BGP routing changes with related path cost changes that occur close in time.
- **Evaluation in an operational network:** We apply our algorithm to routing data collected from a large ISP network, and identify suitable values for the parameters of the algorithm. Our study demonstrates that hot-potato routing is sometimes a significant source of BGP update messages and can cause relatively large delays in forwarding-plane convergence.
- **Exploring the performance implications:** We discuss how hot-potato routing changes can lead to (i) packet loss due to forwarding loops, (ii) significant shifts in routes and the corresponding traffic, and (iii) inaccuracies in measurements of the routing system. We describe how certain operational practices can prevent unnecessary hot-potato routing changes.

These contributions are presented in Sections 3–6, followed by a summary of our results in Section 7.

0. Ignore if egress point unreachable
1. Highest local preference
2. Lowest AS path length
3. Lowest origin type
4. Lowest MED (with same next-hop AS)
5. eBGP-learned over iBGP-learned
6. Lowest IGP path cost to egress point (“Hot potato”)
7. Vendor-dependent tie break

Table 1: Steps in the BGP decision process

2. MODELING HOT-POTATO ROUTING

In this section, we present a precise definition of a “hot potato routing change” and explain why identifying these routing changes in an operational network is surprisingly difficult.

2.1 Hot-Potato BGP Routing Changes

The BGP decision process [1] on a router selects a single best route for each destination prefix by comparing attribute values as shown in Table 1. Two of the steps depend on the IGP information. First, a route is excluded if the BGP next-hop address is not reachable. For example, in Figure 1, the router A does not consider the BGP route from C if A’s forwarding table does not have an entry that matches C’s IP address. Then, after the next five steps in the decision process, the router compares IGP path costs associated with the BGP next-hop addresses and selects the route with the smallest cost—the *closest* egress point. If multiple routes have the same IGP path cost, the router applies additional steps to break the tie. When the BGP decision process comes down to the IGP path cost, we refer to the BGP decision as *hot potato* routing. When a change in an IGP path cost leads a router to select a different best BGP route, we refer to this as a *hot potato routing change*.

To guide our characterization of hot-potato routing, we propose a simple model that captures the path selection process at a single router (which we denote as a *vantage point*):

- **Cost vector (per vantage point)**: The vantage point has a cost vector that represents the cost of the shortest IGP path to every router in the AS. The cost vector is a concise representation of the aspects of the IGP that can affect BGP routing decisions.
- **Egress set (per prefix):** The network has a set of routers that have eBGP-learned routes that are the “best” through step 4 in the BGP decision process. These routes can be propagated to other routers in the AS via iBGP.

For each prefix, the vantage point selects the egress point (from the egress set) with the smallest path cost (from the cost vector). A hot-potato routing change occurs when a vantage point selects a different egress point because of a change in the path cost vector (i.e., that makes the new egress point closer than the old one). For example, initially router A in Figure 1 has an egress set of $\{B, C\}$, path costs of 10 and 9, and a best egress point of C; then, when the path cost to C changes to 11, A selects egress point B. Our goal in this paper is to *determine what fraction of the BGP routing changes are hot-potato routing changes in an operational network*.

2.2 Characterizing Hot-Potato Routing

On the surface, measuring the hot-potato routing changes seems relatively simple: collect BGP and IGP measurements from an operational router and determine which BGP messages were triggered by IGP routing changes. However, several factors conspire to make the problem extremely challenging:

Incomplete measurement data: In a large operational network,

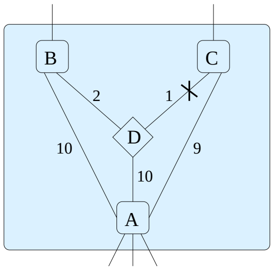


Figure 2: Router A changes best route without path cost change

fully instrumenting all of the routers is not possible; instead, we must work with data from a limited number of vantage points. In addition, commercial routers offer limited opportunities for collecting detailed routing data—we can only collect measurements of the routing protocol messages that the routers exchange amongst themselves. IGP measurements are difficult to collect since they often require a physical connection to a router located in a secure facility. Fortunately, in link-state protocols like OSPF and IS-IS, the routers *flood* the link-state advertisements (LSAs) throughout the network, allowing us to use data collected at one location to reconstruct the path cost changes seen at other routers in the network. However, this reconstruction is not perfect because of delays in propagating the LSA from the point of a link failure or weight change to other routers in the network. Collecting BGP data from multiple routers is easier because BGP sessions run over TCP connections that do not require a physical adjacency. However, BGP messages from the operational router must traverse the network to reach the collection machine, which introduces latency; these delays may increase precisely when the IGP routes are changing. In addition, since BGP is a path-vector protocol, the router sends *only its best route* to its BGP neighbors, making it difficult to know the complete set of routing choices that are available at any given time.

Complex routing protocol dynamics: IGP routing changes stem from topology changes (i.e., equipment going up or down) and configuration changes (i.e., adjustments to the link weights). Monitoring the IGP messages shows only the *effects* of these events. In practice, multiple LSAs may occur close together in time (e.g., the failure of a single router or an optical amplifier could cause several IP links to fail). If one LSA follows close on the heels of another, the routing system does not have time to converge after the first LSA before the next one occurs. Similarly, a prefix may experience multiple BGP routing changes in a short period of time (e.g., a neighboring AS may send multiple updates as part of exploring alternate paths). Similarly, a hot-potato routing change might trigger multiple iBGP routing changes as the network converges. In addition, the global routing system generates a constant churn of BGP updates, due to failures, policy changes, and (perhaps) persistent oscillations. Continuously receiving several updates a second is not uncommon. This makes it difficult to identify which BGP routing changes are caused by hot-potato routing inside the AS. The Multiple Exit Discriminator (MED) attribute introduces additional complexity because the BGP decision process compares MED values only across routes learned from the same next-hop AS, resulting in scenarios where a router's local ranking of two routes may depend on the presence or absence of a third route [11].

Hierarchy of iBGP sessions inside the AS: Large networks often employ *route reflectors* to reduce the overhead of distributing BGP information throughout the AS [12]. However, route reflectors make the dynamics of network-wide routing changes extremely complicated. In the example in Figure 2, router D is a route reflector with clients A, B, and C. Both A and D have shorter IGP paths to C than B. When the C–D link fails, router D shifts its routes from egress C to egress B. However, since A is a client of D, it too would change its routes to use egress B even though its own cost vector has not changed! Determining which BGP routes from A are caused by IGP changes requires knowing the route-reflector configuration of the network and which BGP routing changes from D were caused by the IGP. Some *under-counting* of hot-potato routing changes is inevitable, though focusing the analysis on the "top-level" route reflectors in the network helps limit these effects.

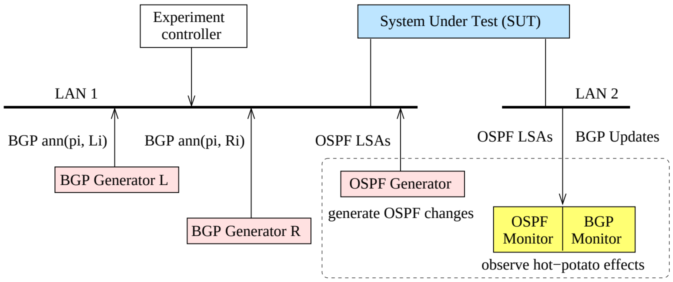


Figure 3: Experimental testbed for router-level testing

Vendor implementation details: Although the routing protocols have been standardized by the IETF, many low-level details depend on implementation decisions and configuration choices. For example, the final tie-breaks in the BGP decision process vary from vendor to vendor. The vendor implementations have numerous timers that control when the router recomputes the IGP paths, reruns the BGP decision process, and sends update messages to BGP neighbors. The router operating system may have complex techniques for scheduling and preempting tasks when multiple events occur close together in time. These router-level details can have a first-order impact on the network-wide dynamics of hot-potato routing.

Together, these issues suggest that computing an exact measure of hot-potato routing changes is extremely difficult, and that we should seek approximate numbers based on reasonable heuristics.

3. CONTROLLED EXPERIMENTS

In this section, we evaluate the dynamics of hot-potato changes on a router in a controlled environment to set the stage for our study of routers in an ISP network. After a description of our testbed, we present a methodology for characterizing the time-average behavior of the router's response to path cost changes. We then present the results of applying this methodology to a Cisco GSR router.

3.1 Router Testbed

The testbed in Figure 3 enables us to perform *black box* experiments on a single router—the *System Under Test* (SUT)—in a controlled fashion. The *OSPF generator* forms an OSPF adjacency with the SUT and sends LSAs to emulate a synthetic topology and to trigger hot-potato routing changes by modifying the link weights. The two *BGP generators* are used to send BGP updates to the SUT. By sending BGP messages with different next-hop IP addresses, the generators can emulate a pair of egress points (L_i and R_i) for each prefix (p_i). The *OSPF monitor* [13] forms an OSPF adjacency with the SUT to log the receipt of LSAs. The *BGP monitor* has an iBGP session with the SUT to observe BGP routing changes. The monitors are software routers that log the protocol messages sent by the SUT. The use of a separate LAN segment isolates these protocol messages from those sent by the OSPF and BGP generators, and allows the two OSPF adjacencies to co-exist.

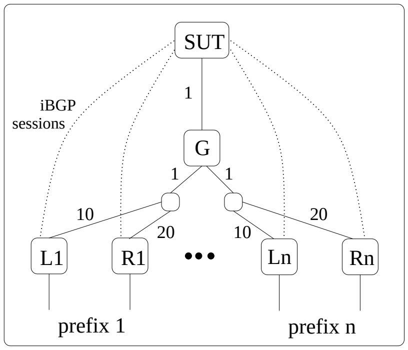


Figure 4: Synthetic network for lab experiments

The *experiment controller* initializes the other machines with the appropriate configuration (e.g., OSPF adjacencies, OSPF link weights, and iBGP sessions). The controller modifies the configuration over time to trigger hot-potato routing changes on the SUT. In practice, we run the OSPF monitor, BGP monitor, and OSPF generator as three processes on the same machine. This ensures that the logging of intradomain topology changes, LSA flooding, and BGP routing changes all share a common time base. Although the timestamp on the OSPF monitor has microsecond resolution, the BGP monitor logs update messages at the one-second level.

3.2 Experiment Methodology

Our experiment is designed to force the SUT to choose between two BGP-learned routes for the same prefix based on the OSPF path cost to the egress point. As shown in Figure 4, the synthetic network has two egress routers, a "left" router L_i and a "right" router R_i, advertising a BGP route for each prefix p_i. The synthetic network has a separate pair of egress routers for each prefix to allow multiple experiments to proceed independently. The two BGP generators send iBGP update messages announcing two BGP routes that differ only in the next-hop attribute—set to the IP address of the corresponding egress router. The OSPF generator acts as the router G and sends LSAs to convince the SUT that the rest of the topology exists. The links from G to the left and right routers have different OSPF link weights—10 and 20, respectively—to control how the SUT selects an egress point[1]. After the BGP sessions and OSPF adjacencies are established, the SUT receives the necessary BGP advertisements and OSPF LSAs to construct the view of the network seen in Figure 4. At the beginning, the SUT selects the route learned from L_i for each prefix p_i, since the path cost of 12 to L_i is smaller than the path of cost 22 to R_i.

After establishing the adjacencies and sending the initial routing messages, we wait for 100 seconds before initiating routing changes, to ensure that the SUT has reached a steady state. In theory, our test could focus on a single destination prefix, triggering repeated hot-potato routing changes over time. However, this approach is problematic for several reasons. First, we would have to ensure that the effects of each OSPF routing change are complete before triggering the next routing change. This would require a tight coupling between the OSPF generator and the two route monitors, or long delays between successive experiments. Second, we would have difficulty conducting truly independently experiments. For example, starting one experiment after the completion of the previous one would not necessarily uncover the time-average behavior of the system. In fact, such an approach might repeatedly observe the system in a particular mode of operation (e.g., a particular part of the timer intervals).

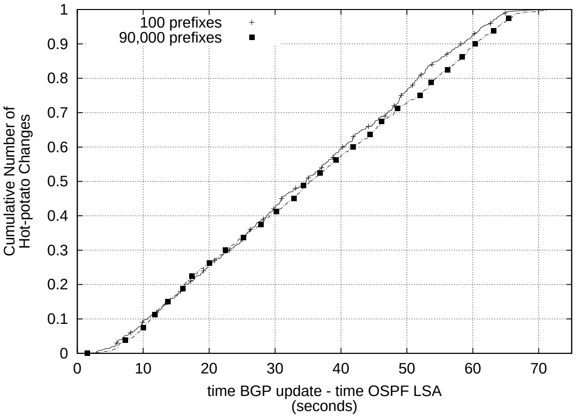


Figure 5: CDF of time lag between OSPF and BGP

Instead, our test proceeds one prefix at a time. The weight on the link to L_i is increased from 10 to 30, for $i = 1, 2, \ldots, n$. Using multiple prefixes obviates the need to estimate an upper bound on the time for any one experiment to complete before starting the next experiment. Instead, we allow for the possibility that the experiment for prefix p_i has not completed before the experiment for prefix p_{i+1} begins. Using multiple prefixes also allows us to evaluate scenarios where multiple OSPF weight changes (affecting different prefixes) occur close together in time. To observe the time-average behavior of the system [14], we impose an interarrival time chosen from an exponential distribution from one prefix to the next. In addition to the test where the link weights change from 10 to 30, we also conduct a test where the link weights decrease from 30 back to 10. Throughout, the OSPF and BGP monitors log the LSAs and BGP updates sent by the SUT. Since each prefix p_i has its own egress routers L_i and R_i, matching an OSPF LSA with the related BGP update message is trivial in this controlled environment.

3.3 Results

Our tests evaluate a Cisco GSR 12012 running IOS 12.0(21)S4 as the SUT. The GSR has a 200 MHz GRP (R5000) CPU and 128 MB of RAM. Although this router has many tunable configuration options, including OSPF and BGP timers, we do not change the values of any tunable parameters and instead focus on the default configuration of the router. The time between the LSA sent by the OSPF generator and the LSA received by the OSPF monitor is less than 30 msec.

Figure 5 shows the cumulative distribution of the time between the OSPF LSA and the BGP update message. The curve marked by "100 prefixes" shows the results for $n = 100$ prefixes and a mean interarrival time of 20 seconds between successive OSPF weight changes across the prefixes. Each run of the test results in 200 experiments—a weight increase and decrease for each of the 100 prefixes—that require about 2.2 hours to complete; the curve represents results for a total of five runs. The curve is almost perfectly linear in the range of 5 to 65 seconds, due to the influence of two timers. First, the router imposes a 5-second delay after receiving

[1] Increasing the number of prefixes and egress routers would create a problem for router G because of the way OSPF generates LSAs. Whenever a link weight changes, the adjacent router sends an LSA with weights of *all* its links, and this LSA must fit in a single packet whose size is constrained by the Maximum Transmission Unit (MTU). Connecting a large number of egress routers directly to G would result in extremely large LSAs that would not fit into a single packet. By having one or more layers of intermediate routers, we keep the fan-out at G (and all other routers) within the limits imposed by the 1500-byte Ethernet MTU.

an LSA before performing the shortest-path computation, to avoid multiple computations when several LSAs arrive in a short period of time [15]. A second LSA that arrives during this interval does not incur the entire five-second delay, as evidenced by the small fraction of LSAs that experienced less than five seconds of delay. Second, the router has a 60-second scan timer that runs periodically to sequence through the BGP routing table and run the BGP decision process for each prefix [16]. The BGP change does not occur until the scan process runs and revisits the BGP routing decision for this prefix. As such, the delay in the BGP routing change is uniform in $[5, 65]$, as evidenced by the straight line in the graph. The Poisson arrival process we use to trigger the OSPF weight changes allows our test to explore the full range of the uniform distribution. A router also imposes a 10-second interval between two consecutive shortest-path calculations, which explains delays in the $[65, 70]$ range.

The second curve ("90,000 prefixes") in Figure 5 shows the results for $90,000$ prefixes. Unlike the "100 prefixes" case, we associate multiple prefixes with every egress router pair. Specifically, we use 100 egress router pairs, and associate 900 prefixes with each pair. Upon a weight change for a given egress pair, the SUT changes the next-hop for all the associated 900 prefixes, and sends out updates for them. The curve plots the results of running the test five times with a mean interarrival time of 20 seconds. Although the "90,000 prefixes" curve looks very similar to the "100 prefixes" curve, the maximum x-axis value for two curves is different—71.34 seconds and 69.78 seconds respectively. This occurs because the scan process is scheduled *after* the previous scan has completed. This makes the interarrival time of the scan process dependent upon the time it takes to run the scan process on the GSR.

Determining which of the many BGP prefixes might be affected by an OSPF path cost change is challenging, which explains why router vendors might choose a timer-driven solution. In practice, many of the timers on the routers are configurable, making it possible to select smaller values that decrease the delay in reacting to hot-potato routing changes, at the expense of higher CPU load. Also, our experiments do not capture the delay for updating the forwarding table with the new best routes; this delay may vary from one router to another. In general, the best choice of a router product and timer settings depends on a variety of complex factors that are beyond the scope of this paper. Still, understanding the router-level timing details is extremely valuable in studying the network-level dynamics of hot-potato routing, as we see in the next two sections.

4. MEASUREMENT METHODOLOGY

In this section, we present our methodology for measuring hot-potato changes experienced by operational routers. Figure 6 presents the steps to correlate BGP updates from a vantage point with OSPF LSAs. (Each dotted box represents steps described in a particular subsection.) Section 4.1 presents the measurement infrastructure used to collect BGP updates and OSPF LSAs. We describe how to compute the path cost vector from the OSPF LSAs in Section 4.2. Section 4.3 explains the classification of BGP routing changes in terms of the possible causes. This sets the stage for the discussion in Section 4.4 about how to associate BGP routing changes with related path cost changes that occur close in time.

4.1 Measurement Infrastructure

We have deployed route monitors running the same software as the monitors described in Section 3.1 in AT&T's tier-1 backbone network (AS 7018). Figure 7 depicts our measurement infrastructure. The OSPF monitor is located at a Point of Presence (PoP) and has a direct physical connection to a router in the network[2]. The monitor timestamps and archives all LSAs. The BGP monitor has iBGP sessions (running over TCP) to several top-level route reflectors. Using an *iBGP* session allows the monitor to see changes in the "egress point" of BGP routes. The BGP monitor also dumps a snapshot of its routes four times a day to provide an initial view of the best route for each prefix for each vantage point. The OSPF and BGP monitors run on two distinct servers and timestamp the routing messages with their own local clocks; to minimize timing discrepancies, both monitors are NTP synchronized.

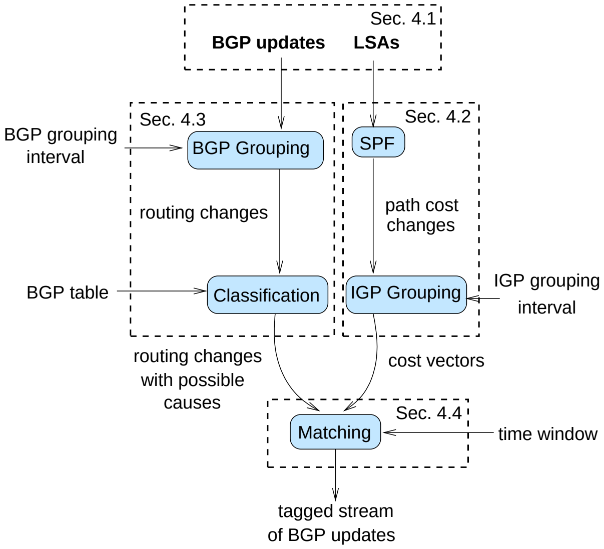


Figure 6: Identifying hot-potato routing changes. Dotted boxes are labeled with the number of the subsection that describes it

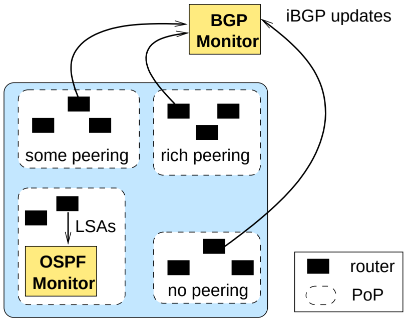


Figure 7: Measurement infrastructure in AT&T backbone

Our analysis focuses on 176 days of data collected from January 2003 to July 2003. Because details of network topology, peering connectivity, and the absolute number of routing messages are proprietary, we omit router locations and normalize most of our numerical results. We study data collected from three vantage points—all Cisco routers that are top-level route reflectors in different PoPs. To explore the effects of router location and connectivity, we select three vantage points in PoPs with different properties. *Rich peering* is a router in a PoP that connects to a large number of peers, including most major ISPs. *Some peering* is a router in a PoP that connects to some but not all major peers. *No peering* is a router in a PoP with no peering connections. Most traffic is directed to

[2] An OSPF network can consist of multiple areas, where area 0 is the "backbone area" that has a complete view of the path costs to reach each router. We connect our monitor to a router in area 0.

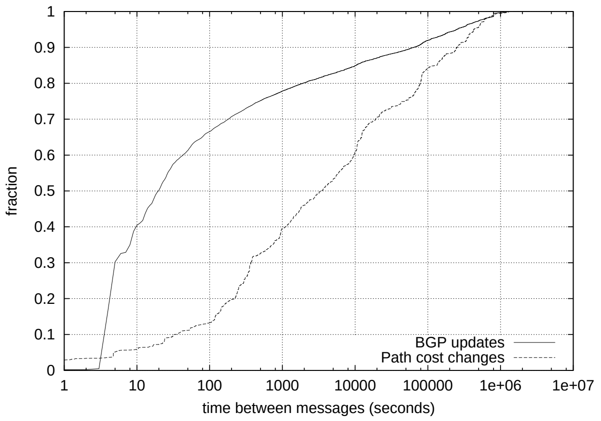


Figure 8: CDF of message interarrivals for each protocol

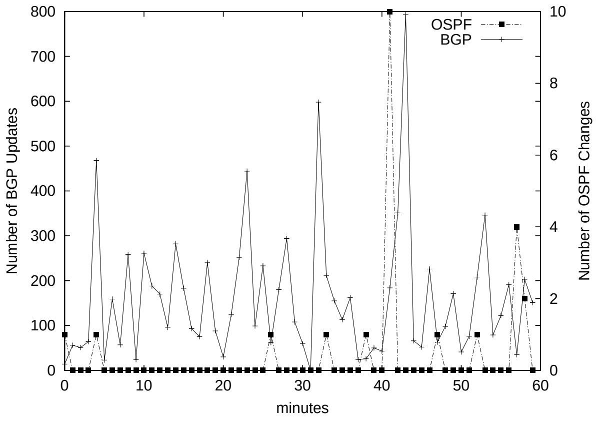


Figure 9: Time series of BGP updates and cost changes

egress points in two nearby PoPs. The three PoPs are located in the eastern part of the United States, relatively close to the location of the two route monitors.

Resets of the monitoring session would affect the accuracy of our results, especially if IGP routing changes are correlated with iBGP session resets. Each of the BGP monitoring sessions experienced at most five resets per month, perhaps due to temporary disruption of the monitor's connection to the rest of the network. These results suggest that IGP events were not a significant contributor to iBGP session resets in the network. In fact, the default keep-alive and hold timers for BGP sessions (60 and 180 seconds, respectively) make it unlikely that transient disruptions during IGP convergence would affect iBGP reachability. Before conducting our analysis, we eliminate all destination prefixes where the BGP routing decisions depend on MEDs; to be conservative, we exclude any prefix that had *any* BGP update with a non-zero MED attribute during the period of the data collection, which represent approximately 13% of the total number of prefixes.

4.2 Computing Cost Vector Changes

OSPF is a link-state routing protocol where each unidirectional link is assigned an administrative weight that is flooded throughout the network in a reliable fashion [2]. Our algorithm processes the LSAs as they arrive to continuously track the OSPF topology and compute the cost vector changes from each vantage point. First, our algorithm disregards any LSAs that do not reflect a change in the OSPF topology; this process excludes OSPF's periodic refresh LSAs as well as any duplicate LSAs sent in the reliable flooding process. For the remaining LSAs, we emulate the OSPF shortest-path computation [2] to determine the path cost from each vantage point to every other router at the boundary of the network (i.e., any router that could serve as an egress point for one or more prefixes).

Some OSPF topology changes do not trigger path cost changes. For example, some links with high OSPF weights do not appear on any shortest path (e.g., links under maintenance or provisioning); an increase in the weight or the failure of the link would not affect any of the shortest paths. Also, some links always appear as part of multiple shortest paths along with other links (e.g., parallel links between two routers). Other LSAs may change the path costs for one vantage point but not another. Whenever an LSA changes one or more path costs for a given vantage point, our algorithm produces a new cost vector for that vantage point. If the vantage point cannot reach another router (e.g., due to a failure or network partition), we represent the path cost as ∞. Our study focuses on the common case of cost changes from one finite value to another.

In practice, multiple LSAs may occur close together in time. Even if these LSAs stem from different events (e.g., two independent failures), the delays in propagating the LSAs and in converging to new routes make it impossible to analyze these LSAs separately. Instead, we group path-cost changes that occur within a small time window into a single cost vector change. We select the interval duration based on analysis of our OSPF measurements, shown by the "path cost changes" curve in Figure 8. To generate the curve, we consider the interarrival times of the path cost changes between each vantage point and all possible egress routers and plot the resulting cumulative distribution. About 5% of the path cost changes occur within ten seconds of each other. These may correspond to LSAs caused by a single physical event, such as rebooting a router. Otherwise, the curve increases gradually over the range of values. Half of the path cost changes have an interarrival time of more than 3400 seconds, and 10% are more than 252,000 seconds (almost a month). In the next Section, we apply a time interval of 10 seconds for grouping path cost changes; additional experiments showed that the results were not sensitive to small changes in the size of the interval.

4.3 Classifying BGP Routing Changes

The global BGP routing system generates a continuous stream of update messages, as shown by the example in Figure 9. This graph plots the number of BGP updates (left y-axis) and path cost changes (right y-axis) seen by the "rich peering" router over one hour, with one-minute bins. In this example, the router sometimes makes several hundred BGP routing changes in a minute. In contrast, very few intervals have more than a handful of path cost changes, and these changes do not necessarily cause the router to switch from one egress point to another for any prefix. The large volume of BGP updates stems, in part, from the exploration of multiple alternate routes when a router switches from one best path to another [17, 8]. These short-lived BGP routes do not correspond to stable path changes but rather the *transition* from one stable route to another. The details of path exploration depend on timing details at routers throughout the Internet. Instead, in our study, we are interested in how IGP cost changes cause a router inside the AS to switch from one stable route to another with a different egress point.

To focus on changes from one stable route to another, we group BGP updates at the same router for the same prefix that occur close together in time, based on the "BGP updates" curve in Figure 8. To generate the curve, we consider the interarrival times of the BGP updates from each vantage point for each prefix and plot the resulting cumulative distribution. More than 30% of the BGP updates

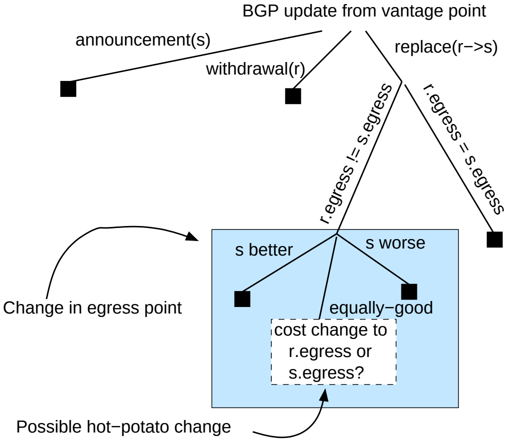


Figure 10: Classifying BGP routing changes at a vantage point

have an interarrival time of five seconds or less. This stems from the 5-second minimum-route advertisement timer used by Cisco routers to pace the update messages on iBGP sessions. Previous studies have shown that interarrival times of around 30 seconds are quite common for external routing changes, since Cisco routers use a 30-second minimum-route advertisement timer for eBGP sessions [17]. In Figure 8 about two-thirds of the BGP updates have a spacing of less than 70 seconds. In the next Section, we apply a time interval of 70 seconds for grouping BGP messages to combine many of the transient updates occurring during path exploration. Additional experiments showed that the results were not sensitive to small changes in the size of the grouping interval.

Many BGP routing changes have no relationship to the cost vector changes in the interior of the network. Drawing on the BGP decision process, our algorithm classifies BGP routing changes in terms of their possible causes. Starting with an initial BGP routing table, we consider a stream of changes in the best route for each prefix. Figure 10 illustrates how we classify a BGP routing change from route r to route s for a prefix at a particular vantage point. Hot-potato routing changes cause a router to switch from one BGP route to another. As such, changing from or to a null route does not represent a hot-potato routing change. However, hot-potato routing changes can cause s to *replace* r. In this case, further analysis helps narrow down the possible causes. If r and s have the same egress point, a change in the cost vector cannot be responsible.

Having different egress points $r.egress$ and $s.egress$ does not necessarily imply that hot-potato routing is responsible. The new route s might be "better" than the old one at some earlier stage in the decision process; for example, s might have a shorter AS path or a larger local-preference. Alternatively, the route r might have been withdrawn; because our monitor sees only the best route at each vantage point, we can only infer that r was withdrawn if s is "worse" than r. Hence, if r and s are not "equally good" through steps 0–5 of the BGP decision process, we can dismiss hot-potato routing as a possible cause. If the routes are equally good, hot-potato routing *might* be responsible if the relative "closeness" of the two egress points has changed—making the egress point s closer than egress point r.

4.4 Matching Cost Changes with BGP

To further refine our inference that an IGP routing change caused the vantage point to select s, we inspect the stream of cost vectors for this vantage point to see if $s.egress$ became closer than $r.egress$ within some small time interval. We verified the correctness of this algorithm using the testbed presented in Section 3. In this scenario, all BGP routes are stable and the only changes are related to path cost changes; our algorithm correctly identified the OSPF LSA that caused each BGP update. However, BGP routes are *not* stable in the operational network. Hence, our algorithm might mistakenly match a BGP routing change with an *unrelated* cost vector change. The BGP routing change might have been triggered by an external event, such as a policy change or a failure in another AS, that caused r to be withdrawn or replaced by a less attractive route. Yet, a seemingly-related cost vector change could occur nearby in time that is consistent with the vantage point's decision to switch to route s. In this situation, our algorithm would mistakenly associate the replacement of r by s with the cost change. (In practice, the IGP event might have caused a similar BGP routing change anyway if the external event had not happened first!)

Although these kinds of mismatches are difficult to avoid completely, three aspects of our algorithm reduce the likelihood of false matches: (i) preprocessing the cost vector changes and BGP update messages as discussed in Section 4.2 and 4.3, (ii) the fine-grained classification in Figure 10 which eliminates many of the external BGP routing changes, and (iii) the careful selection of the time window for correlating the two datasets. To find the appropriate time window, we first consider cost vector changes that occur within ten minutes before or after the BGP routing change. Although our algorithm did find occasional matches over the entire 20-minute interval, the vast majority of hot-potato BGP routing changes occurred within *three minutes* of the cost vector change, for reasons we explore in more detail in the next section. In experiments where we did *not* preprocess the OSPF and BGP data, we tended to see a larger number of (presumably false) matches in the large time intervals, suggesting that our preprocessing is helpful for reducing the likelihood of false matches.

Our algorithm finds some matches where the BGP routing change appears to happen 1–2 seconds *before* the cost vector change. Although this seems counter-intuitive, this can occur in practice for two reasons. First, the OSPF LSA may take longer to reach our OSPF monitor than for the related BGP update to reach the BGP monitor. The reliable flooding of OSPF LSAs is typically implemented in software on the router, which may subject these messages to higher delays. In contrast, BGP update messages are sent via a TCP connection between two routers; the IP packets carrying these messages traverse the hardware forwarding path through the routers. Second, the BGP monitor has a coarser timestamp resolution than the OSPF monitor. To account for these two issues, we allow a small *negative* time difference between the cost vector change and the BGP change. Therefore, we believe a time window of $(-2, 180)$ is a reasonable way to avoid false matches while still capturing the bulk of the real hot-potato routing changes. We use this window for the analysis in the rest of the paper.

5. HOT POTATOES IN THE WILD

This section presents a case study of hot-potato routing changes in an operational network. Our goal is to identify and understand the main properties of hot-potato routing changes, rather than to highlight specific numerical values that might vary from one network to another. Although most hot-potato routing changes occur within 60 seconds, extra delays of 1–2 minutes sometimes arise due to the iBGP hierarchy and the transfer of update messages. The frequency of hot-potato routing changes varies significantly across time and router location. Interestingly, the hot-potato BGP updates have a much more even spread across the destination prefixes than the remaining update messages.

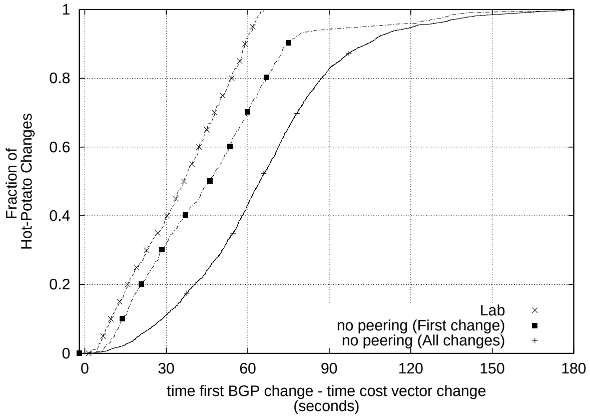


Figure 11: CDF of time lag between the cost vector change and related BGP routing changes, using a 10-second window to group OSPF LSAs, a 70-second window to group the BGP update messages, and a $(-2, 180)$ window to correlate the cost vector changes with BGP routing changes.

5.1 BGP Reaction Time to Cost Changes

Figure 11 presents the cumulative distribution of the delay between a cost vector change and a correlated BGP routing change for the "no peering" router from January 2003 to July 2003. The graph shows a significant gap between the results for the lab experiments and the curve for *all* hot-potato changes sent by the "no peering" router. Upon receiving a new LSA, the router must (i) rerun the IGP shortest-path computation, (ii) apply the BGP decision process to select the best route for each prefix, and (iii) send update messages to BGP neighbors for the routes that have changed. The first two steps represent the time required to react to a cost vector change, and the third step depends on the number of BGP routing changes. Our lab experiments in Section 3 evaluated only the first two steps in a controlled environment. In order to have a fair comparison, we measure the delay between the cost vector change and the *first* prefix experiencing a hot-potato routing change.

The graph shows that most hot-potato routing changes occur within 80 seconds of the cost vector change, which is closer to the 70 seconds upper limit of our controlled experiments. The extra 10 seconds are explained by the rate of LSA arrivals and the number of routes in an operational router. When the rate of LSAs is higher, the likelihood of incurring the 10-second delay between consecutive shortest-path calculations as explained in Section 3.3 is increased. The scan process may require several seconds in an operational router because of the large number of BGP routes. The 60-second timer restarts after the *completion* of the previous scan; hence, the BGP reaction time also includes the time for the running time of the scan process. These two factors contribute to longer reaction times in the operational router. We discuss the reaction times longer than 80 seconds in the next subsection.

5.2 Transfer Delay for Multiple Prefixes

The difference between the curve for *all* hot-potato changes and the one for the *first* change corresponds to the delay to transfer BGP updates for multiple prefixes. When a cost vector change affects a large number of prefixes, the transmission of the BGP update messages to iBGP and eBGP neighbors introduces additional delay, as shown in Figure 12. This graph highlights two specific cost vector changes that affected the *largest* number of prefixes for the "no peering" and "some peering" routers during June 2003. Although the BGP update for the *first* prefix appears within 80 seconds of the path vector change, some updates appear much later. For example, in the "no peering" curve, a single cost vector change affected the BGP routes for more than $80,000$ prefixes. Although the BGP change for the first prefix occurs 66 seconds after the cost vector change, the routing change for the last prefix occurred 83 seconds later, 149 seconds after the OSPF change.

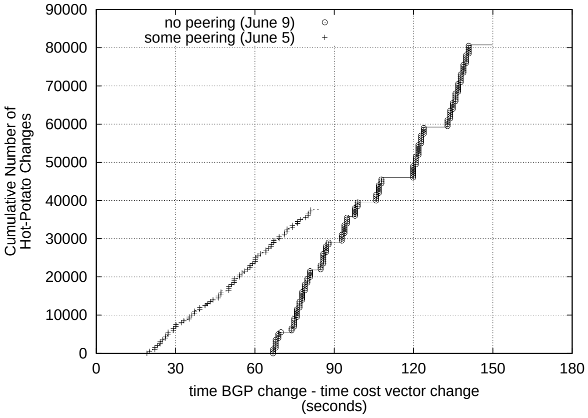


Figure 12: BGP transfers caused by one cost vector change

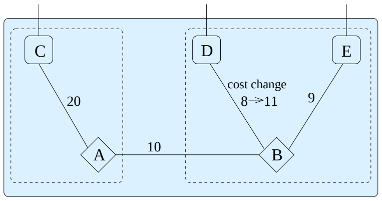


Figure 13: Router A waits for B's decision.

The shape of this curve is mainly determined by the volume of data and the TCP transmission rate between the vantage point and the BGP monitor. In our experiments, the BGP monitor is within a few hundred miles of the "no peering" router and the update packets travel just a few hops through the network. Longer delays might be possible over iBGP sessions between pairs of routers with longer round-trip times, which may also contribute to longer delays in reacting to hot-potato routing changes. The "no peering" curve has some gaps that are 3 to 4 seconds long. These gaps are caused by the minimum-route advertisement timer, which limits the rate of BGP updates in a session. The smaller steps (one second long) are due to the one-second granularity of the BGP monitor timestamp.

The transfer delay may also be responsible for the instances in Figure 11 in which the reaction time exceeds 80 seconds for the "first change" curve. These kinds of delays may be caused by the propagation of hot-potato BGP routing changes from one router to another, as shown in Figure 13. In the example, routers A and B are route reflectors and routers C, D, and E are egress points; C is a client of A, and D and E are clients of B. Initially, A and B select egress point D, with path costs of 18 and 8, respectively. A is unaware of the route via E because B only advertises its best route to A. When the B-D cost increases to 11:

1. The LSA is flooded throughout the network and each router computes new path costs to D. For example, A and B compute new path costs of 21 and 11, respectively.
2. After their scan timers elapse, A and B rerun the BGP decision process. If A runs first, A selects the egress point C with a path

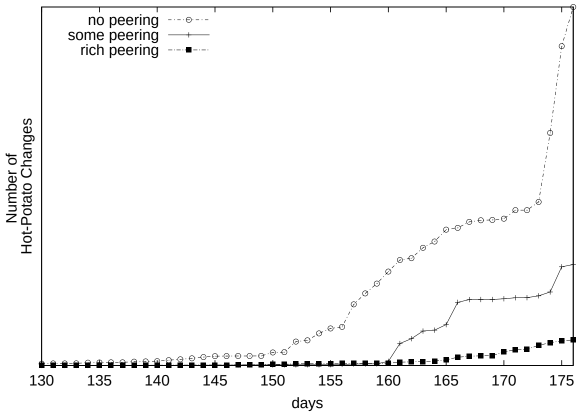


Figure 14: Hot-potato changes across days and locations

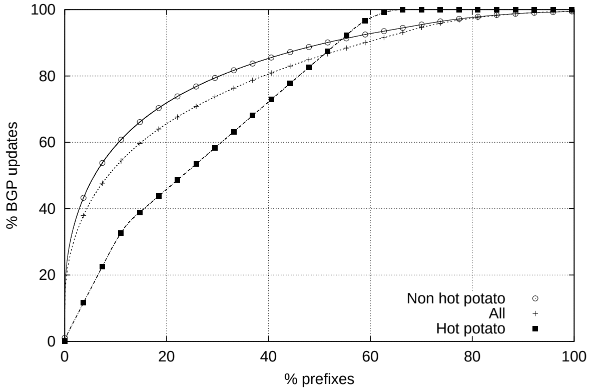


Figure 15: CDF of BGP updates across destination prefixes

cost of 20, since this is smaller than 21. Sometime afterwards, B selects egress point E.

3. B sends the new route (with egress point E) to A, and A selects egress point E with a path cost of 19.

Suppose a cost vector change triggers a large number of BGP updates from B, but some of these updates do not trigger hot-potato changes in A. Then, A may have to wait for the transfer of a number of BGP updates before experiencing a hot-potato change. This explains some of the reaction times longer than 80 seconds in Figure 11. Other instances with longer reaction times may also be due to false matches in associating a BGP routing change with a cost vector changes. Cost vector changes for which BGP takes more than 80 seconds to react trigger 807 BGP routing changes on average, whereas those that have smaller reaction times trigger 3398 BGP updates on average.

Combining the results of the "first change" curve in Figure 11 and the transfer delays in Figure 12, a router's reaction to cost vector changes may take 0–80 seconds for the first prefix and an additional 80 seconds (in extreme cases) for the remaining prefixes. Combining these effects, the vast majority of hot-potato changes take place within three minutes of the cost vector change, as is shown in the "all changes" curve in Figure 11.

5.3 Temporal and Spatial Variability

The influence of hot-potato routing varies significantly across time. Figure 14 presents the number[3] of hot-potato updates. For ease of presentation, the graph plots the days in increasing order of the number of hot-potato BGP routing changes and we only show the 46 days with higher number of hot-potato changes. The plot shows that on most days the routers did not experience *any* hot-potato routing changes. Still, on a few days the number was much higher. For the "no peering" router, one day had an unusually large number of hot-potato routing changes that were responsible for 82% of the BGP routing changes on that day. The variability across the days may stem from natural differences in the time and location of IGP weight changes and maintenance activity. The large variation across days makes it difficult to define a representative statistic for the frequency of hot-potato routing changes.

Comparing the three curves in Figure 14 highlights the influence of the location of the router on the likelihood of hot-potato routing changes. Over the period of our study, the "rich peering" router was always the least affected by path cost changes, as seen by the bottom curve lying very close to the x-axis in Figure 14. The likelihood that a path cost change affects the selection of the BGP best route depends on the proximity of the router to each of its nearby egress points. For the "rich peering" router, many of the prefixes have an egress point at the same PoP. Very few path cost changes would cause the router to select a different egress point for these prefixes. This suggests that a natural way to reduce the number of hot-potato routing changes would be to have rich peering at *every* PoP. However, having rich peering at all locations is infeasible in practice, due to cost and geographic constraints. A service provider is bound to have routers in some remote locations that are not close to PoPs of the other large providers.

5.4 Hot-Potato Variation Across Prefixes

Previous studies have shown that a small fraction of unstable prefixes are responsible for most of the BGP route updates [7, 8, 9] The BGP routes for the remaining prefixes stay the same for days or weeks at a time. Figure 15 plots the cumulative distribution of BGP update messages across the destination prefixes for the "no peering" router for June 2003. To compare our results with previous work, the graph plots the number of *BGP update messages* rather than the number of BGP routing changes. The prefixes are sorted according to their contribution to the number of BGP messages. The middle curve corresponds to all of the BGP messages. About 20% of the prefixes contribute 65% of the BGP updates, consistent with previous findings. However, the bottom curve shows that the distribution of BGP updates caused by *hot-potato* routing changes has a much more even spread across the prefixes.

The broader distribution across prefixes occurs because cost vector changes can affect the path costs to reach the egress points for a wide variety of prefixes. Still, some prefixes do not experience *any* hot-potato BGP updates, as seen in the flat portion in the upper-right part of the graph. This corresponds to prefixes with a very small number of egress points, including the prefixes that have a *single* egress point. Every router in the network would always pick this single egress point as the best egress point for the prefix. Still, the relatively uniform distribution across the remaining prefixes may have important implications. For prefixes that generally have stable *eBGP*-learned routes, internal path cost changes could be a primary cause of the BGP routing changes observed inside an AS. Since some of these prefixes may be responsible for a large volume of traffic, limiting the frequency of hot-potato routing changes may be useful to avoid large traffic shifts and transient performance disruptions.

[3] Although the graph omits the values on the y-axis, the three curves are plotted in proportion starting at $y = 0$.

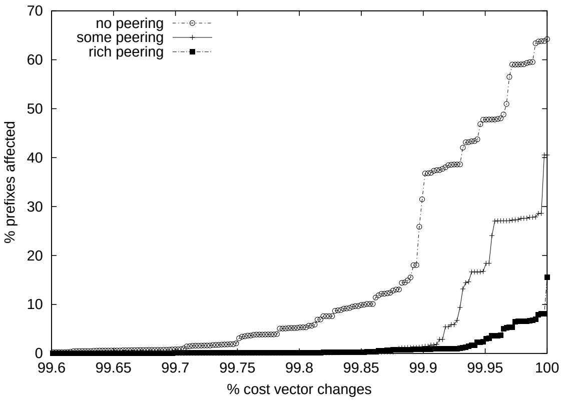


Figure 16: Fraction of prefixes affected by cost vector change

6. IMPLICATIONS OF HOT POTATOES

Hot-potato changes in BGP routing influence network performance by causing shifts in the flow of traffic to neighboring domains and extra delays in the convergence of the forwarding plane. In addition, hot-potato changes can introduce inaccuracy in active measurements of the forwarding plane and external monitoring of BGP update messages. Certain operational practices for network design and engineering can limit these effects.

6.1 Performance Degradation

6.1.1 Routing and Traffic Shifts

Hot-potato routing can sometimes cause a router to change the egress points for multiple destination prefixes, which could lead to significant congestion on certain paths through the Internet. In Figure 16, we explore how many destination prefixes are affected at a single router when a path-cost change occurs. More than 99% of the path-cost changes do not affect the egress point for any prefix. The vast majority of intradomain events occur far away from the router, and as such do not affect the path costs for nearby egress points. Even when changes occur closer to the router, they might not affect the router's local ranking of the two closest egress points for a given prefix. However, when hot-potato routing changes *do* occur, the effects can be dramatic. For the "no peering" router in the top curve in Figure 16, 0.1% of the path-cost changes affect the BGP route for more than 40% of the prefixes.

These kinds of routing changes can lead to sudden increases in traffic at the new egress points and along the downstream paths. For an estimate of these effects, we computed the average traffic volume for each destination prefix using Netflow measurements from the periphery of the network. The prefixes affected by the hot-potato routing changes in Figure 16 account for 5–35% of the traffic in the network. This gives a preliminary indication that some hot-potato routing changes cause significant shifts in traffic, though a more detailed study is necessary to understand the relationship between hot-potato routing changes and traffic volumes.

6.1.2 Slow Forwarding-Plane Convergence

Compared to other kinds of routing changes, hot-potato routing changes cause longer delays in forwarding-plane convergence, since each router must recompute its IGP routes *and* rerun the BGP decision process before updating the forwarding table. Differences in when the routers revisit their BGP decisions can lead to transient forwarding loops, as illustrated in Figure 17. In this example, the AS has four routers and two egress points to prefix d. The numbers on the edges represent the IGP link weights, and we omit the full-mesh of iBGP sessions for simplicity. At first, routers B and C both identify router A as the closest egress point, causing C to direct traffic to d through B. When the weight of the B–A link increases to 111, both routers eventually switch to the route learned at D. However, if B runs its BGP decision process first and updates its forwarding table, B starts forwarding traffic destined to d toward D while C continues to forward the traffic toward A—resulting in a forwarding loop.

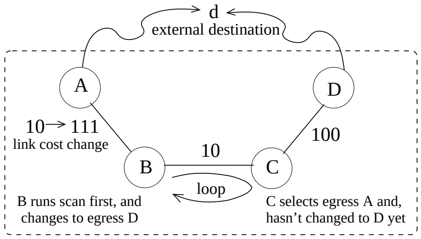


Figure 17: Transient forwarding loop for packets destined to d

During the interval before C runs its decision process and updates its forwarding-table entry for d, all packets destined to d are caught in a forwarding loop between B and C. The packets would repeatedly traverse the loop until the IP Time-to-Live (TTL) field expires, causing one of the routers to discard the packet. The forwarding loop causes packet loss for the hosts communicating with d, and increased congestion for other traffic traversing the B–C link. Depending on the alignment of the BGP scan timers on the two routers, this problem can persist for up to 60 seconds, even though the intradomain routing protocol has converged[4]. If TCP transfer latency or the iBGP hierarchy cause larger delays in forwarding-plane convergence, the loops can persist even longer.

According to a previous study of packet-level measurements in a large ISP backbone [18], most forwarding loops last for less than 10 seconds. This is consistent with typical delays for IGP convergence [6, 19]. However, the study also found that, for one of the links, about 35% of the loops persisted for 10–60 seconds. Based on our results, we speculate that these forwarding loops can be explained by hot-potato routing changes.

6.2 Measurement Inaccuracies

6.2.1 Active Probes of the Forwarding Plane

The effects of slow forwarding-plane convergence may be difficult to capture using traditional active measurement techniques. Service providers and third-party measurement companies deploy probe machines in various parts of the network in order to exercise the paths between pairs of hosts. Referring to Figure 17, suppose the provider connected one probe machine to router A and another to router D. Probe packets sent from A to D would traverse the path A–B–C–D. When the IGP weight of the B–A link changes, these probes might experience temporary loss while the IGP reconverges. However, the forwarding path of the probe packets would *not* be affected by the 60-second scan timer since there would be no change in the egress point used to reach the destination address of the probe packets; both B and C continue to use the egress point

[4] Note that the extra convergence delay for hot-potato routing changes does *not* affect the stability of the forwarding path for the iBGP sessions themselves. The IP packets sent over iBGP sessions travel between routers within the backbone and the forwarding of traffic between these routers depends only on the IGP! The delivery of BGP updates to our route monitor is not affected either, since the network has a single egress point to reach the monitor.

D to reach the destination probe machine. This is true, in general, for probe machines that connect to a single location inside an AS. As such, measurements between these kinds of probe machines would only capture the transient effects of *IGP* convergence, and not the combined *IGP-BGP* convergence process. Accurately capturing the performance impact of hot-potato routing changes would require a more complex active measurement infrastructure with probe machines reachable through multiple egress points.

6.2.2 External Analysis of BGP Updates

A hot-potato routing change does not necessarily cause an AS to advertise new BGP routes to neighboring ASes. First, the export policy for the eBGP session might filter the route. This decision depends on the commercial relationship with the neighbor (e.g., a route learned from one peer would not be exported to another) and on whether route aggregation is performed. Second, the router might decline to forward the new route if it does not differ significantly from the old route. For example, routers typically perform *non-transitive attribute filtering* to avoid propagating routes that differ only in local attributes (like BGP next-hop or local-preference) rather than global ones (such as AS path). Third, the router might not propagate the route due to BGP timers, such as the minimum-route advertisement timer, that are used to pace the rate of updates to neighboring ASes. If the router changes its best BGP route for the prefix multiple times during the interval, the intermediate BGP routes would not be visible to the neighbor.

For a rough estimate of the externally-visible updates, we look at BGP routing changes that affect the *AS path attribute*, since these would be propagated to neighboring domains subject to the export policy and the BGP timers. Referring back to the example in Figure 1, A switches egress points without a change in the AS path; we would not classify this routing change as externally visible. However, if router B connected to a *different* next-hop AS with a path to the destination, the AS path would change; router A would propagate the new route to its eBGP neighbors. Looking over the month of June, we estimate that around 14% of the hot-potato routing changes seen at the "no peering" router would be sent to its neighbors; this would account for 5% of the externally-visible BGP routing changes. For the "some peering" router, these two numbers are 5% and 2%, respectively—about 60% smaller than for the "no peering" router. Although these average numbers are relatively small, the values vary substantially from day to day; on one day *all* hot-potato updates at all three routers had changes in the AS path.

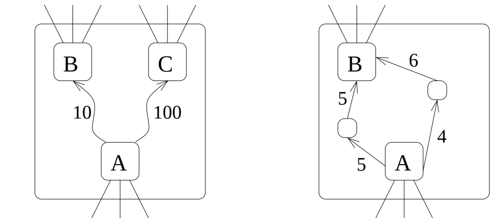
These externally-visible BGP updates may affect the results of research studies based on public BGP routing data [20, 21] collected from eBGP sessions with routers in large ASes throughout the Internet. Depending on which router in an ISP network connects to these public servers, the contribution of hot-potato routing changes to the data may vary significantly! For example, a hot-potato routing change that affects a large number of prefixes in one network may be indistinguishable from a BGP session reset at another nearby location, when viewed from outside the AS.

6.3 Recommended Operational Practices

Avoiding hot-potato routing changes helps prevent shifts in traffic, extra delays in forwarding-plane convergence, and externally-visible BGP updates. This can improve the end-to-end performance of Internet traffic flowing through the AS.

6.3.1 Selection of IGP Path Costs to Egress Points

Comparing the results for the "rich peering" and "no peering" routers shows how much the design of the network affects the prevalence of hot-potato routing changes. This suggests two main ways to reduce the likelihood of these kinds of routing changes. First, each router should have a *single nearest egress point* for reaching most destination prefixes. As shown in Figure 18(a), router A has a small IGP path cost of 10 to reach egress point B and a much larger IGP path cost of 100 to reach C. This reduces the likelihood that small variations in IGP path costs would trigger a hot-potato routing change at A. Only a very dramatic internal network event would cause A to choose egress point C over egress point B. Second, each router should have *two or more shortest IGP paths* to the nearby egress point. As shown in Figure 18(b), router A has two shortest paths (with an IGP path cost of 10) to egress point B. This decreases the likelihood that a single internal event would change the IGP path cost to reach B and, as such, would tend to avoid hot-potato changes in the BGP routes. Having multiple shortest paths between pairs of routers is also useful to reduce the latency for forwarding-plane convergence for IGP routing changes [22], even when no BGP-level change occurs.



(a) Router A closer to B than C (b) Router A with two shortest paths

Figure 18: Preventing hot-potato routing changes

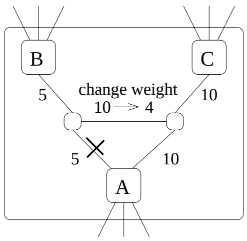


Figure 19: A still picks egress B during maintenance

6.3.2 Traffic Engineering and Planned Maintenance

Operators tune the IGP link weights to adapt the flow of traffic through the network in response to network congestion and planned maintenance [23]. For example, suppose the operators need to upgrade the operating system on a router. Before disabling the router, the operators may adjust the IGP weights on other links in the network to prevent congestion during the maintenance period. Operators can take the effects of hot-potato routing into account when making changes to the IGP configuration[5]. For example, in Figure 19 the router A selects egress point B with an IGP path cost of 10 over egress C with a cost 20. However, if the left link from A needs to be disabled for upgrading, the path cost to B would increase to 25, making C the closer egress point. The hot-potato routing change can be avoided by changing the weight of the middle link from 10 to 4 before the maintenance activity; this ensures that the path to B has cost 19—smaller than the cost to C.

Despite the many benefits, completely avoiding hot-potato rout-

[5] A route emulation tool can be used to model the effects of IGP changes on the flow of traffic [23]. Avoiding hot-potato routing changes obviates the need to collect detailed per-prefix traffic statistics as input to the tool; in the absence of egress-point changes, a simple router-to-router traffic matrix would suffice.

ing changes may be impossible, or conflict with other engineering goals. Still, operators can try to avoid violating the guidelines when designing the network topology and selecting IGP weights.

7. CONCLUSIONS

The interplay between intradomain and interdomain routing has important implications on the stability and efficiency of Internet routing and, in turn, on end-to-end performance. In this paper, we presented a methodology for joint analysis of OSPF and BGP measurement data and a characterization of the interplay between the protocols in an operational network. Our results suggest that hot-potato routing may play an important role in BGP routing changes, and that BGP updates can lag 60 seconds (or more!) behind the related IGP events. This can lead to surprisingly long delays in forwarding-plane convergence that greatly exceed the typical delays for IGP convergence [6, 19]. We also show that the number of hot-potato routing changes varies significantly across time and router location, suggesting a need for further analysis and modeling of how the protocols interact. Our ongoing work focuses on:

Performance implications: To better understand the significance of hot-potato routing changes, we plan to perform a detailed analysis of shifts in traffic, forwarding loops, and externally-visible BGP updates. We are also conducting additional lab experiments to evaluate the trade-off between fast convergence and router CPU load. Finally, we are exploring protocol extensions and operational practices that decrease the sensitivity of BGP decisions to IGP changes.

Extensions to matching algorithm: We are studying refinements to our heuristic for identifying hot-potato changes. One heuristic is to check that the *first* BGP update caused by a cost vector change occurs within the first 80 seconds, even if the remaining updates take longer to appear. Another heuristic is to check for hot-potato changes in all the prefixes that have the same set of egress points.

Conservative algorithm: We are developing a conservative algorithm for identifying hot-potato routing changes. The main idea is to collect iBGP routing data from a much larger number of vantage points to track the evolution of (part of) the egress set over time. When a router switches from one egress point to another, we check if the new route has existed for a while and the old route continues to exist. This provides a conservative way to identify hot-potato routing changes, without requiring a separate feed of IGP data.

Detailed hot-potato model: We are working on a detailed model of hot-potato routing changes that captures how the iBGP hierarchy affects the routing choices available to each router. We plan to formalize the insights from our measurement results about the influence of router location, peering connectivity, and network topology, on the likelihood of hot-potato routing changes, and the influence of routing protocol timers on convergence delay.

Acknowledgments

We would like to thank Jay Borkenhagen, Nick Feamster, Flavio Junqueira, Rich Kwapniewski, Zhuoqing Morley Mao, Dan Pei, Jia Wang, and the anonymous reviewers for their comments. Thanks also to Alex Gerber for his help with the Netflow data, Olaf Maennel for his help in debugging our lab testbed, and Matt Roughan for suggesting we use multiple prefixes in our lab experiments.

8. REFERENCES

[1] "A Border Gateway Protocol 4 (BGP-4)." Internet Draft draft-ietf-idr-bgp4-23.txt, work in progress, November 2003.

[2] J. Moy, "OSPF Version 2." RFC 2328, April 1998.

[3] R. Callon, "Use of OSI IS–IS for Routing in TCP/IP and Dual Environments." RFC 1195, December 1990.

[4] D. Watson, C. Labovitz, and F. Jahanian, "Experiences with monitoring OSPF on a regional service provider network," in *Proc. International Conference on Distributed Computing Systems*, pp. 204–213, May 2003.

[5] A. Shaikh, C. Isett, A. Greenberg, M. Roughan, and J. Gottlieb, "A Case Study of OSPF Behavior in a Large Enterprise Network," in *Proc. Internet Measurement Workshop*, November 2002.

[6] G. Iannaccone, C.-N. Chuah, R. Mortier, S. Bhattacharyya, and C. Diot, "Analysis of link failures in an IP backbone," in *Proc. Internet Measurement Workshop*, November 2002.

[7] C. Labovitz, A. Ahuja, and F. Jahanian, "Experimental Study of Internet Stability and Wide-Area Network Failures," in *Proc. Fault-Tolerant Computing Symposium*, June 1999.

[8] J. Rexford, J. Wang, Z. Xiao, and Y. Zhang, "BGP Routing Stability of Popular Destinations," in *Proc. Internet Measurement Workshop*, November 2002.

[9] S. Agarwal, C.-N. Chuah, S. Bhattacharyya, and C. Diot, "Impact of BGP Dynamics on Intra-Domain Traffic," in *Proc. ACM SIGMETRICS*, June 2004.

[10] C. Labovitz, R. Malan, and F. Jahanian, "Internet Routing Instability," *IEEE/ACM Trans. Networking*, vol. 6, pp. 515–558, October 1998.

[11] T. G. Griffin and G. Wilfong, "An analysis of the MED oscillation problem in BGP," in *Proc. International Conference on Network Protocols*, 2002.

[12] S. Halabi and D. McPherson, *Internet Routing Architectures*. Cisco Press, second ed., 2001.

[13] A. Shaikh and A. Greenberg, "OSPF monitoring: Architecture, design, and deployment experience," in *Proc. Symposium on Networked Systems Design and Implementation*, March 2004.

[14] R. Wolff, "Poisson arrivals see time averages," *Operations Research*, vol. 30, no. 2, pp. 223–231, 1982.

[15] "Configure Router Calculation Timers." `http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/12cgcr/%np1_c/1cprt1/1cospf.html#xtocid2712621`.

[16] Understanding BGP Processes on Cisco. `http://www.cisco.com/warp/public/459/highcpu-bgp.html#topic1`.

[17] C. Labovitz, A. Ahuja, A. Bose, and F. Jahanian, "Delayed Internet routing convergence," *IEEE/ACM Trans. Networking*, vol. 9, pp. 293–306, June 2001.

[18] U. Hengartner, S. B. Moon, R. Mortier, and C. Diot, "Detection and analysis of routing loops in packet traces," in *Proc. Internet Measurement Workshop*, November 2002.

[19] C. Alaettinoglu, V. Jacobson, and H. Yu, "Toward milli-second IGP convergence." Expired Internet Draft, draft-alaettinoglu-isis-convergence-00.txt, November 2000.

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

[21] "RIPE NCC RIS." `http://www.ripe.net/ripencc/pub-services/np/ris-index.html`.

[22] A. Sridharan, S. B. Moon, and C. Diot, "On the correlation between route dynamics and routing loops," in *Proc. Internet Measurement Conference*, October 2003.

[23] B. Fortz, J. Rexford, and M. Thorup, "Traffic engineering with traditional IP routing protocols," *IEEE Communication Magazine*, October 2002.