Destination Reachability and BGP Convergence Time

Beichuan Zhang
University of California, Los Angeles
bzhang@cs.ucla.edu

Daniel Massey
Colorado State University
massey@cs.colostate.edu

Lixia Zhang
University of California, Los Angeles
lixia@cs.ucla.edu

***Abstract*— One important performance measure for routing protocols is packet delivery. An ideal routing protocol should quickly adapt to topological changes and deliver packets as long as any path to the destination exists. In this paper, we examine the packet delivery performance in a network running the BGP routing protocol when a destination may be disconnected from time to time. We develop two metrics, *extra downtime* and *false uptime*, to capture the time difference between actual loss of connectivity and perceived unreachability. Our results show that extra downtime closely matches T_{up} convergence delay, and false uptime closely matches T_{down} convergence delay. Furthermore, our results show that, for transient connectivity failures, a shorter T_{down} convergence time can have negative impact on packet delivery.**

I. Introduction

The primary goal of a routing protocol is to *deliver* packets. Other routing protocol performance metrics, such as routing stability and convergence delay, are all important issues, however these measures themselves should also be considered with respect to maximizing packet delivery. Ideally, a routing protocol should deliver packets to destinations as long as any valid path exists that can reach the destination. In today's Internet, network failures, caused by either planned maintenance or unexpected events, occur frequently [1][2]. Border Gateway Protocol (BGP [3]), the de-facto inter-domain routing protocol, can adapt to failures by converging to a new set of valid paths. However, the routing adjustment may take a long time due to various delays in the propagation of update messages and the exploration of alternative paths. As a result, the period of destination *unreachability* can be substantially longer than the time period of actual physical *connectivity* losses.

There have been a plethora of BGP performance studies in recent years; a number of them focused on BGP's convergence behavior. Labovitz *et al.* [4] categorizes BGP routing events into four basic types: T_{down} (a previously reachable destination is withdrawn), T_{up} (a previously unreachable destination is announced), T_{long} (an existing path is replaced by a longer one), and T_{short} (an existing path is replaced by a shorter one). They observed that T_{up} and T_{short} events typically converge in a relatively short time period, but T_{down} and T_{long} events can trigger path explorations and take several minutes or more to converge. Previous efforts, such as [5][6][7], proposed various techniques to improve BGP convergence time without considering the impact on packet delivery. As shown by Pei *et al.* [8], packet delivery performance is related, but not equivalent, to the routing convergence time.

In this paper we use BGP as a case study to evaluate packet delivery performance in reaction to a loss of connectivity followed by a recovery (i.e., a T_{down} followed by a T_{up}). For a given period of *physical* disconnectivity to a destination, we measure the length of routing induced unreachability periods perceived by data sources. Due to route propagation and convergence delay, the source may not be aware that the destination becomes unreachable until some time later, and if the source sends packets during this time, those packets will be dropped in the network after consuming some amount of network resources. Similarly, the downtime observed by a source can be longer than the actual disconnectivity period. We define two new metrics: *false uptime* and *extra downtime*, to measure the resource overhead and the reachability lost, respectively.

Our results show that, generally speaking, false uptime matches closely to T_{down} convergence delay, and extra downtime matches closely to T_{up} convergence delay. Thus reducing convergence time can indeed improve packet delivery. Our study supports an earlier claim [9] that BGP's long MRAI[1] timer adds substantial delay to routing convergence, suggesting a re-examination of the MRAI timer value in order to further improve packet delivery. However such a re-examination is not simple and in fact, for transient failures, a shorter T_{down} convergence time may lead to *reduced* packet delivery. To maximize packet delivery, a routing protocol must not only reduce convergence delay but must also mask transient failures.

II. Packet Delivery and Destination Reachability

Given a source $\mathcal{S}$ and a destination $\mathcal{D}$, we define the following concepts:

Definition** 1:* $\mathcal{D}$ is ***connected at time t if there is at least one physical path leading from $\mathcal{S}$ to $\mathcal{D}$. Otherwise $\mathcal{D}$ is ***disconnected*** at time t.

Definition** 2:* $\mathcal{D}$ is ***reachable from $\mathcal{S}$ at time t if a packet sent by $\mathcal{S}$ at time t will eventually be delivered to $\mathcal{D}$. Otherwise $\mathcal{D}$ is ***unreachable*** from $\mathcal{S}$ at time t.

The reachability to $\mathcal{D}$ at time t cannot be measured by examining the routing tables at all the routers at time t, because it takes time for a packet to propagate through the network and the routing tables may change during this time period.

[1]Minimum Route Announcement Interval, which is used to space out consecutive route announcement messages.

Instead, the reachability to $\mathcal{D}$ is measured in the following way. Suppose $\mathcal{S}$ sends a packet at time t_0, and the link delay to the next hop router n_1 is l_1, then the packet will arrive at n_1 at time $t_1 = t_0 + l_1$. n_1 forwards the packet based on its forwarding table at time t_1. Assuming the link delay to the next hop router n_2 is l_2, n_2 will receive the packet at time $t_2 = t_1 + l_2$ and forwards the packet based on its forwarding table at t_2. If there are k routers along the path from $\mathcal{S}$ to $\mathcal{D}$, $\mathcal{D}$ is considered *reachable* from $\mathcal{S}$ at time t if and only if router n_i at time $t_{i-1} + l_i$ has a forwarding entry for $\mathcal{D}$ that points to n_{i+1}.

One can see from the above that reachability is not equivalent to connectedness. Connectedness is determined by the physical connectivity in the topology at time t, while reachability at time t is determined by the summation of per hop forwarding state in sequential time order starting from time t. For example, during routing convergence, although a physical path to $\mathcal{D}$ may exist, $\mathcal{D}$ may still be unreachable from source $\mathcal{S}$ due to stale routing state at certain routers along the path. Similarly, even though a physical path to $\mathcal{D}$ may not exist at time t, a packet sent at that time may still be delivered to $\mathcal{D}$ if physical connectivity is re-established while the packet is making its way towards $\mathcal{D}$.

Let $disconnected(s)$ denote the number of seconds that $\mathcal{D}$ is physically disconnected from $\mathcal{S}$. If $\mathcal{D}$ is disconnected multiple times, $disconnected(s)$ is the sum of all the individual time periods when $\mathcal{D}$ is disconnected. Similarly, let $downtime(s)$ denote the number of seconds that $\mathcal{D}$ is unreachable from $\mathcal{S}$. If the reachability is intermittent, $downtime(s)$ is the sum of all the time periods when $\mathcal{D}$ is unreachable.

We define the *extra downtime* perceived by $\mathcal{S}$ as:

$$\varepsilon(s) = downtime(s) - disconnected(s)$$

$\varepsilon(s)$ measures the reachability lost due to routing behaviors such as convergence delay. By definition, the destination itself always has zero extra downtime: $\varepsilon(d) = 0$.

$downtime(s)$ can be further divided into two parts: *no-path* time and *false uptime*. During no-path time, $\mathcal{S}$ does not have a path to $\mathcal{D}$ and thus does not attempt to send any packets. During false uptime, $\mathcal{S}$ believes it has a path to $\mathcal{D}$, however packets sent during this time period will not reach $\mathcal{D}$. False uptime is caused by routing convergence delay after the destination has lost its connectivity. We denote false uptime as $f(s)$. From the network's point of view, all packets sent during $f(s)$ travel one or more hops before eventually getting dropped, resulting in wasted network resources.

III. Analysis

Since we focus on BGP at the inter-domain level, both $\mathcal{D}$ and $\mathcal{S}$ represent Autonomous Systems (AS) in our analysis. We assume that a failure happens within $\mathcal{D}$ at time t_{d1}, and then recovers at time t_{d2}, disconnecting the destination host from the rest of the Internet for a total amount of time $u = t_{d2} - t_{d1}$. Once the failure/recovery is detected, BGP takes some time to propagate the new information and converge on new routes.

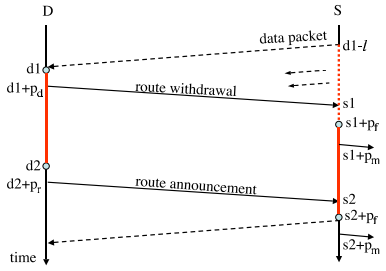


Fig. 1. Downtime Analysis (ideal convergence)

To establish basic concepts for our analysis, we first consider an "ideal convergence" scenario to derive formulas for both extra downtime and false uptime, and then extend our results by considering realistic BGP convergence behavior.

A. Ideal Convergence

In an ideal case, after a failure, a BGP router never selects an invalid path nor delays updates by MRAI. Therefore, routing convergence time is determined solely by the physical limits of propagation and processing delay along the best path. Under these assumptions, we will show that extra downtime and false uptime are given as follows.

$$\varepsilon(s) = t_{up}(s) + d(s) \tag{1}$$

$$f(s) = t_{down}(s) + d(s) \tag{2}$$

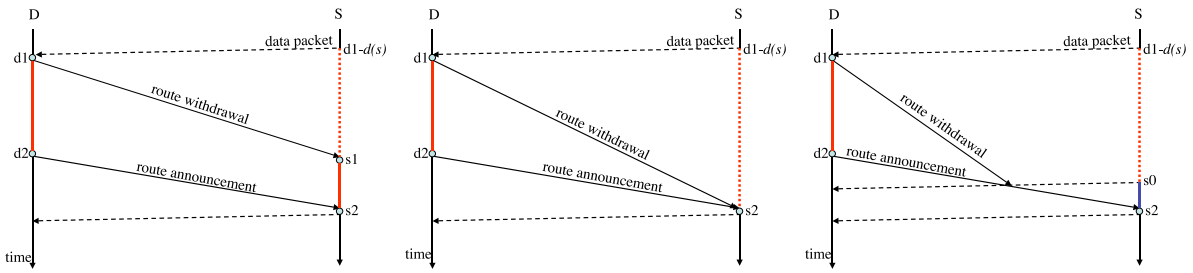
where $d(s)$ is the propagation delay from $\mathcal{S}$ to $\mathcal{D}$, $t_{down}(s)$ is the convergence time of the T_{down} event, and $t_{up}(s)$ is the convergence time of the T_{up} event.

Figure 1 illustrates the ideal convergence behavior. The T_{down} event begins when $\mathcal{D}$ encounters a failure at time t_{d1}. After a delay of p_d, which includes failure detection time and message processing time, and etc., $\mathcal{D}$ sends out a route withdrawal. This withdrawal message incurs a link delay of l before arriving at $\mathcal{S}$. $\mathcal{S}$ takes time of p_f to re-calculate the best path and update its forwarding table, in this case, to withdraw the path to $\mathcal{D}$. It also takes processing time of p_m before $\mathcal{S}$ sends this withdrawal further to its neighbor. Since $\mathcal{S}$ updates its forwarding table at time $t_{s1} + p_f$, the convergence time for this T_{down} event is $t_{down}(s) = (t_{s1} + p_f) - t_{d1} = p_d + l + p_f$.

False uptime occurs due to propagation and processing delays. The propagation delay from $\mathcal{S}$ to $\mathcal{D}$ is $d(s) = l$, and thus all packets sent at or after $t_{d1} - l$ will encounter the failure and be dropped. However, $\mathcal{S}$ does not stop forwarding packets until time $t_{s1} + p_f$ when it updates the forwarding table. This results in false uptime $f(s)$ of:

$$f(s) = (t_{s1} + p_f) - (t_{d1} - l) = p_f + p_d + 2l = t_{down}(s) + d(s)$$

The T_{up} event begins when $\mathcal{D}$ recovers from the failure at time t_{d2}. After a delay of p_r, which includes detection time, processing time and etc., $\mathcal{D}$ announces the recovered path to its neighbors. $\mathcal{S}$ receives the announcement after a link delay of l. Again, $\mathcal{S}$ takes time of p_f to update the forwarding table, and



(a) $t_{down}(s) < t_{up}(s) + u$ (b) $t_{down}(s) = t_{up}(s) + u$ (c) $t_{down}(s) > t_{up}(s) + u$

Fig. 2. Downtime Analysis (slow convergence)

p_m to forward this announcement to its neighbors. Since the forwarding table is updated at $t_{s2} + p_f$, the convergence time for this T_{up} event is $t_{up}(s) = (t_{s2} + p_f) - t_{d2} = p_r + l + p_f$.

Since the destination is disconnected from t_{d1} to t_{d2}, $disconnected(s) = u = t_{d2} - t_{d1}$. As noted above, packet delivery begins to fail at time $t_{d1} - l$, and $\mathcal{S}$ does not resume packet forwarding until time $t_{s2} + p_f$. All packets sent after time $t_{s2} + p_f$ will reach the destination. Therefore $downtime(s) = (t_{s2} + p_f) - (t_{d1} - l)$ and the extra downtime is

$$\varepsilon(s) = downtime(s) - u = p_r + p_f + 2l = t_{up}(s) + d(s)$$

The results easily generalize to when $\mathcal{S}$ is m hops away from $\mathcal{D}$. Consider that at each intermediate hop, the routing update is delayed by $l + p_m$. For a source m hops away from the destination, $t_{down}(s) = p_d + (m-1)(l + p_m) + (l + p_f)$, $t_{up}(s) = p_r + (m-1)(l + p_m) + (l + p_f)$. Using these we find that Equations 1 and 2 still hold. Note that in this ideal case since $t_{down}(s)$, $t_{up}(s)$, and $d(s)$ are all linearly proportional to m, overall both extra downtime and false uptime are linearly proportional to the distance between the source and the destination.

B. BGP Route Convergence

In reality, after a failure, BGP updates can be delayed due to MRAI, and a router may explore some obsolete paths before choosing the new best path. MRAI, and to a lesser extent path exploration, also impact the convergence behavior after connectivity is restored. As a result the convergence time of $t_{down}(s)$ and $t_{up}(s)$ may increase and become more difficult to analyze. It may also happen that the T_{down} event is still in progress when it is overtaken by the subsequent T_{up} event. Figure 2 illustrates three possible scenarios.

In Figure 2(a), T_{down} convergence ends before T_{up} ends. More precisely, $t_{down}(s) < t_{up}(s) + u$. In this case, Equations 1 and 2 still hold. The MRAI timer and path exploration can lead to longer T_{down} and T_{up} convergence time, which directly translates into longer false uptime and extra downtime, respectively.

In Figure 2(b), $t_{down}(s) = t_{up}(s) + u$, which means that the T_{down} convergence ends exactly at the same time as the recovery information arrives. In this case, $\mathcal{S}$ does not ever withdraw its route to $\mathcal{D}$, and it keeps forwarding packets. During the time period between $t_{d1} - d(s)$ and t_{s2}, all the packets are dropped in the network, and as a result, $\mathcal{S}$ experiences the maximum possible false uptime of $t_{up}(s) + u + d(s)$.

In Figure 2(c), the T_{down} convergence is still in progress when the (path recovery) T_{up} updates overcome the T_{down} updates. In this case, the actual value of $t_{down}(s)$ is unknown since it is cut short by the new announcements, but we can conclude that $t_{down}(s) > t_{up}(s) + u$. Not only does $\mathcal{S}$ keep forwarding packets, the packets sent between t_{s0} and t_{s2}, though not guaranteed, may follow some fluctuating path and reach $\mathcal{D}$. In the extreme case where a T_{down} is followed quickly by a T_{up} and $\mathcal{D}$ never sends out the withdrawal, $\varepsilon(s) = d(s)$ and $f(s) = u + d(s)$. Overall we have:

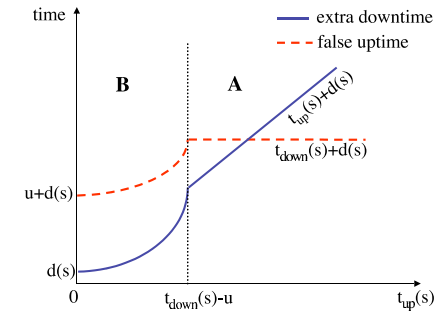
$$d(s) \leq \varepsilon(s) \leq t_{up}(s) + d(s) \tag{3}$$

$$u + d(s) \leq f(s) \leq t_{up}(s) + u + d(s) \tag{4}$$

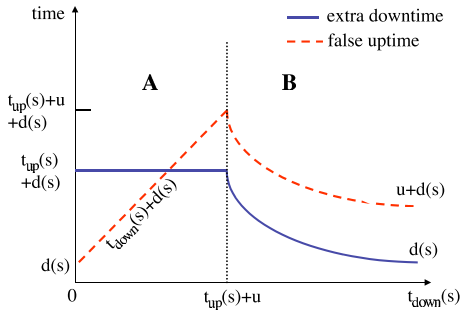
Figure 3 summarizes the trends of extra downtime and false uptime. Each figure is divided into two areas, area A when $t_{down}(s) < t_{up}(s) + u$, and area B when $t_{down}(s) > t_{up}(s) + u$. Equations 1 and 2 hold in area A, while 3 and 4 hold in area B.

Most BGP convergence improvement proposals reduce T_{down} convergence time, but have very little effect on T_{up} convergence time [10]. Therefore, Figure 3(b) can be used to explain how these proposals may impact reachability metrics. If we are already in area A applying these convergence improvements, thus reducing $t_{down}(s)$, would have no impact on how many packets are delivered (i.e., same $\varepsilon(s)$), but would reduce the network resources wasted during convergence period (i.e., shorter $f(s)$). However, if we are already in area B, reducing $t_{down}(s)$ would increase $\varepsilon(s)$, leading to *reduced packet delivery*. Intuitively, when $t_{down}(s) > t_{up}(s) + u$, the source does not remove the route to the destination, thus it avoids a possibly lengthy T_{up} convergence and has more packets delivered than with shorter $t_{down}(s)$.

We cannot quantify $\varepsilon(s)$ or $f(s)$ solely based on analysis. For example, in Figure 2(c), the number of packets delivered



(a) t_{up}



(b) t_{down}

Fig. 3. Reachability and Convergence Time

between t_{s0} and t_{s2} depends on transient paths in the network. Therefore, we use simulations to verify the analysis results.

IV. Simulation

We model BGP as a Simple Path Vector Protocol [5], and conduct simulations using SSFNET [11]. We use a binary tree topology and a 110-node topology derived from the Internet inter-domain graph [12]. Each node in the topology represents an AS; in the rest of the paper we use the words *node* and *AS* in an interchangeable way. All the nodes contain data sources. A destination $\mathcal{D}$ is attached to a randomly selected AS in the topology. We initialize each simulation run by letting every node establish a stable route to $\mathcal{D}$ before taking measurements. After the initial route establishment, the destination AS sends a withdrawal message to all its neighbors signaling a connectivity failure to $\mathcal{D}$. After u seconds, the failure is repaired, and the destination AS sends an announcement signaling the recovery. The simulation ends when all the nodes establish a stable path to reach $\mathcal{D}$ again. In this scenario, $\mathcal{D}$'s duration of disconnectivity is the same for all the source nodes, i.e., $disconnected(s) = u$, the failure duration.

In a previous work [8], destination reachability was measured by sending packets to the destination and counting the percentage of packets received. However the percentage of packet delivery is determined not only by the routing behavior (the behavior we want to measure), but also by the packet injection rate (an arbitrary value chosen as part of the experiment). In other words, changing the packet injection rate can lead to different packet delivery results for the exact same routing dynamics. Because the packet injection rate in real networks is unknown and likely to vary, we take a different and more accurate approach in this work.

We directly calculate the destination reachability from the forwarding tables using the method described in Section II. We record the forwarding tables of all the nodes over the entire convergence period, from when the destination AS withdraws its route until all the nodes establish stable paths to the destination again. We then derive the actual *forwarding path* each packet takes by taking into account link delays, and calculate how long each forwarding path lasts based on the propagation of routing update messages. Each packet is either delivered to the destination or dropped because it encounters a node which does not have a route to the destination or because its TTL value becomes zero. For simplicity, we do not consider queuing delay at each node.

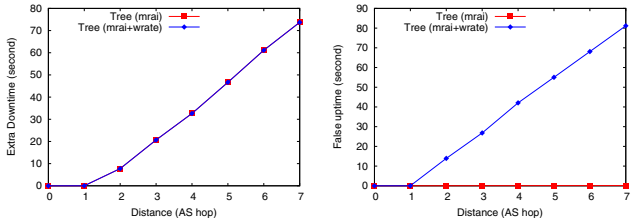
During routing convergence period, transient routing loops can occur [13]. Packets encountering loops may be dropped due to TTL expiration, or continue on the forwarding path after the loop is resolved. We set the TTL in each packet to the maximum value of 255 and decrease the TTL by one at each node. Note that in practice, the TTL may start with a lower value and is decremented at each *router* as opposed to each AS. Our simulation setting is a conservative approach to looping losses, and our results show that using different initial TTL values have negligible effect on the values of extra downtime and false uptime.

In simulations, we set each link delay to 10 ms, MRAI to 30 seconds, and the failure duration to 60 seconds unless otherwise stated. Although the BGP specification states that MRAI should be applied on a per *destination-prefix* basis, vendor implementations typically apply MRAI on a *per-peer* basis, which means that a routing update for one destination may be delayed by MRAI due to earlier update messages for other destinations. Our simulation imitates a per-peer MRAI setting, and each routing update experiences a delay at each hop ranging from 0 to 30 seconds.

A. Impact of MRAI

To examine the impact of MRAI, we use a binary tree topology with 127 nodes. Because there is only a single path from any source to any destination in a tree topology, no path exploration can occur during convergence. In such a topology, we expect that $t_{up}(s) \cong \overline{mrai} \cdot m$, where $\overline{mrai}$ is the average delay due to MRAI at each AS hop, and m is the number of AS hops from the source to the destination. By default MRAI is not applied to route withdrawals, thus we expect $t_{down}(s) \cong m \cdot l$. For comparison, we also run simulations with the WRATE option turned on. WRATE (Withdrawal RATE limiting) applies MRAI to withdrawals, which increases $t_{down}(s)$ to $\overline{mrai} \cdot m$.

The simulation results are shown in Figure 4. Each data point is the average over all the nodes that have the same



(a) Extra Downtime (b) False Uptime

Fig. 4. MRAI Simulation Using Tree Topology

distance to the destination. Because routing updates propagate along a single path to all the other nodes, $t_{down} < t_{up} + u$, thus the analytical results from area *A* in Figures 3(a) and 3(b) apply. Equations 1 and 2 predict that both $\varepsilon(s)$ and $f(s)$ should be linear to m. This fits well with the simulation results. By delaying withdrawal messages, WRATE increases t_{down} from tens of milliseconds to tens of seconds but does not affect t_{up}. Equations 1 and 2 predict that WRATE should have a major impact on false uptime, but almost no impact on extra downtime. These predications are confirmed by the simulation results.

B. The Impact of Path Exploration

To examine the impact of path exploration, we run simulations with a 110-node Internet-derived topology [12] with rich connectivity among the nodes. As a results, path exploration tends to occur after a connectivity failure which can delay routing convergence. Figure 5 shows the simulation results with different parameter settings: BGP with and without policy, different failure durations, and BGP with convergence improvement.

1) BGP with $u = 60s$: As an AS's distance from the destination increases, T_{down} convergence time increases. When the distance is greater than 3 AS hops, the condition $t_{down}(s) < t_{up}(s) + u$ no longer holds true. At roughly 3 AS hops, we reach the analytical turning point between area *A* and area *B* in Figure 3. The simulation results in Figure 5 show an initial linear increase in the extra downtime as t_{up} increases and then become much flatter when the distance is greater than 3 AS hops. After 3 AS hops, $t_{down}(s) > t_{up}(s) + u$ and area *B* of Figure 3(b) shows how extra downtime become smaller than the linear extrapolation.

2) BGP with $u = 960s$: Figure 3(b) shows that, failure duration u affects the turning point $t_{up}(s) + u$ and the upper bound of false uptime $t_{up}(s) + u + d(s)$. When the failure duration is increased from $u = 60s$ to $u = 960s$, $t_{down}(s) < t_{up}(s) + u$, thus the analytical behavior described by area *A* of Figure 3 applies. As expected, the results in Figure 5 show that the extra downtime becomes roughly linear to the AS hop count. With the upper bound of false uptime raised, $f(s)$ also becomes significantly higher, reflecting the long t_{down} caused by path exploration.

3) BGP with Policy: Routing policy can impact convergence delay by reducing the number of paths that can be explored during routing convergence. Routing policies are often used to enforce commercial relationship between ASes by regulating the selection and announcement of BGP paths. Even when a physical path exists between two ASes, it will not be considered for best-route selection or be announced to other ASes if a policy forbids doing so. We simulated a routing policy derived from common operational practice to observe how it affects reachability.

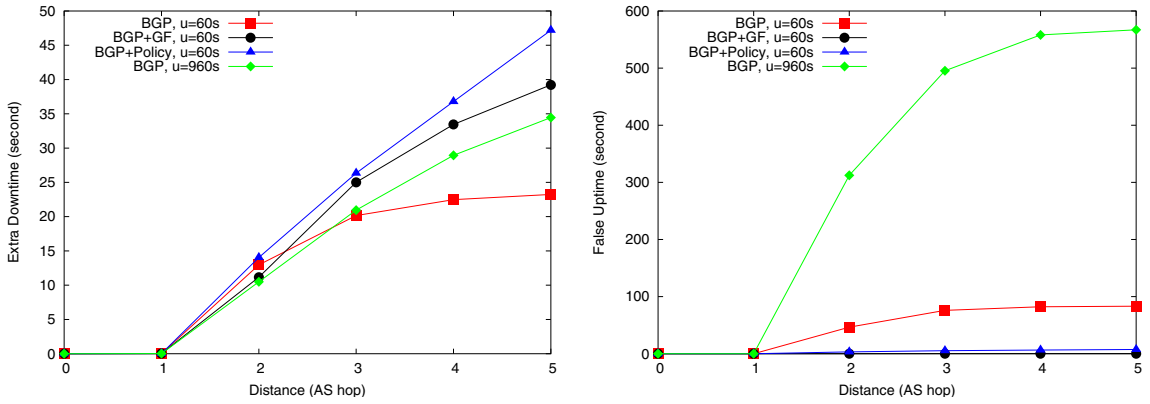
In the 110-node Internet-derived topology, we choose the top seven nodes with highest degree to form a tier-1 set. Nodes that directly connect to tier-1 form tier-2, nodes that directly connect to tier-2 form tier-3, and so on. Nodes at the same tier are "peers", nodes at higher tier are "providers" to nodes at lower tier, which are "customers". Policy dictates that node x does not forward data traffic to neighbor y unless the traffic is either from x's customers or destined to y's customers. This policy is often referred to as the "NoValley" policy [14], and is believed to be widely adopted in practice.

Because the routing policy reduces alternative paths that can be explored during routing convergence, the T_{down} convergence time is reduced so that $t_{down}(s) < t_{up}(s) + u$. Our analytical results predict downtime to be linear with the number of AS hops, and small false uptime, which match the simulation results in Figure 5. In fact, the NoValley policy has reduced alternate paths to the degree that the remaining paths together look very close to a tree, leading to a strongly linear result similar to that of a binary tree topology in Figure 4.

C. The Impact of Convergence Improvement

We conclude our simulation results by considering the impact of proposed BGP convergence enhancements. We selected one convergence enhancement, Ghost Flushing (BGP-GF) [6], and simulated its behavior in the 110-node Internet-derived topology. With BGP-GF, T_{down} convergence time becomes very small, thus $t_{down}(s) < t_{up}(s) + u$. We expect extra downtime becomes almost linear to the number of AS hops, and false uptime becomes very small, which is confirmed by Figure 5. The results for BGP-GF are generally representative of the other convergence enhancements. Pei *et al.* [10] shows that various BGP convergence enhancements share a common behavior of reducing T_{down} convergence time while having little impact on T_{up} convergence time.

When the path to the destination is withdrawn from a source, it may take some time to restore the path after the destination is connected again. If a long T_{down} convergence time delays the arrival of path withdrawal which is then caught up by the following T_{up} announcement, the source can continue packet forwarding, thus eliminating extra downtime by skipping T_{up} convergence time, but at the cost of bandwidth overhead due to the long false uptime. Whether BGP-GF improves overall packet delivery depends on the tradeoff between the above two factors. Without BGP-GF, $t_{down}(s)$ can be either smaller or greater than $t_{up}(s) + u$. In the former case, BGP-GF's only effect will be to reduce $f(s)$. In the latter case, BGP-GF will



(a) Extra Downtime

(b) False Uptime

Fig. 5. Simulation Results in an Internet-derived Topology

reduce $f(s)$, but increase εs. The reduction of $f(s)$ highly depends on u. When u is small, the overhead caused by false uptime is small, thus not withdrawing the path becomes a better choice because of shorter $\varepsilon(s)$. In this case, applying the current convergence enhancements may have *negative impact* on overall reachability.

The failure duration u plays an important role in determining whether the convergence enhancements will have negative effect on reachability. We are not aware of any data on the failure duration at the inter-domain level. At the intra-domain level, Iannaccone *et al.* [1] reported that in a Tier-1 provider network, failures happen frequently, and 40% failures last less than one minute, and 80% failures last less than 15 minutes. Internet measurement results in [4] and [15] show that T_{down} convergence time can be as many as several minutes longer than T_{up} convergence time. These suggest that the case of $t_{down}(s) > t_{up}(s) + u$ may indeed occur in the operational Internet.

One alternate approach is to develop techniques that reduce T_{up} convergence time. This will reduce the extra downtime even if the source has removed its path. In theory, reducing MRAI will reduce T_{up} convergence time. However, since MRAI is an important mechanism in BGP to limit update sending rate, any change to it is likely to affect other BGP behaviors (e.g., damping), and requires careful research. Another approach is for the destination not to send a withdrawal for transient failures. However this raises the challenge of how to estimate the expected duration of a failure.

V. Related Work

Labovitz *et al.* investigated BGP convergence time in [4][9]. They observed that T_{down} and T_{up} events converge much slower than T_{up} and T_{long} events. Analysis shows that during routing convergence period, path vector protocols such as BGP can potentially explore a large number of alternate paths, many of which are obsolete, resulting in excessive updates and long convergence time. Many solutions have been proposed to reduce BGP slow convergence. BGP-Assertion [5] enforces route consistency checking before accepting a path. Ghost Flushing [6] uses "flushing withdrawal" to accelerate the removal of obsolete paths from the network. RCN [7], FESN [16], and RCO [17] attach the "root cause" of a routing event to update messages, so that routers can differentiate stable paths from obsolete paths. All these work focus on BGP's routing behavior only, e.g., convergence time and update message overhead; they lack the study on implications to reachability or packet delivery.

The work most related to ours is [8]. Pei *et al.* compared various routing protocols in packet delivery performance during a single T_{long} event. In T_{long}, the failure happens in the middle of the network, and the destination is connected throughout the entire event. Our work defines new metrics to measure reachability, and studies BGP packet delivery during a composite event, a T_{down} followed by a T_{up}.

Internet packet delivery performance and network failures have been widely studied, e.g., [18]. Most previous work use packet count as performance metric to study the end-to-end packet delivery performance, which is the combined result of many factors. Our work uses new metrics and focuses only on inter-domain routing, in order to gain deeper understanding of one important factor individually. We recognize that other factors such as intra-domain routing play important roles to the end-to-end performance too, and plan to study them in the future work.

VI. Conclusion

In this paper we use BGP as a case study to examine packet delivery performance in the reaction to simple topological events of a loss of connectivity followed by a recovery (i.e., T_{down} followed by T_{up}). For a given loss of connectivity to the destination, we use *extra downtime* and *false uptime* to measure the difference between the actual failure duration and

the perceived reachability by the rest of the network provided by the routing protocol. Our results show that, although extra downtime closely matches T_{up} convergence delay and false uptime closely matches T_{down} convergence delay, when failure durations are short, shorter T_{down} convergence time may actually lead to *reduced* packet delivery. Possible solutions include minimizing T_{up} convergence delay, which requires a reduction of MRAI value in the case of BGP, and masking transient failures from the rest of the network. Our work demonstrates the importance of considering packet delivery measures in improving routing protocol performance.

VII. Acknowledgments

We would like to thank Dan Pei, Xiaoliang Zhao, and Andreas Terzis for their insightful discussions and comments. We are also grateful to anonymous reviewers for their helpful feedback. This work was partially supported by NSF grant ANI-0221453 and by a research grant from Cisco Systems.

References

[1] G. Iannaccone, C.-N. Chuah, R. Mortier, S. Bhattacharyya, and C. Diot, "Analysis of link failures over an IP backbone," in *ACM SIGCOMM Internet Measurement Workshop (IMW)*, Nov. 2002.

[2] C. Labovitz, A. Ahuja, and F. Jahanian, "Experimental study of internet stability and wide-area network failures," in *Proceedings of FTCS99*, June 1999.

[3] Y. Rekhter and T. Li, "Border Gateway Protocol 4," Internet Engineering Task Force, RFC 1771, July 1995.

[4] C. Labovitz, A. Ahuja, A. Abose, and F. Jahanian, "Delayed internet routing convergence," in *Proc. of ACM SIGCOMM*, 2000.

[5] D. Pei, X. Zhao, L. Wang, D. Massey, A. Mankin, S. F. Wu, and L. Zhang, "Improving bgp convergence through consistency assertions," in *Proc. of IEEE INFOCOM*, 2002.

[6] A. Bremler-Barr, Y. Afek, and S. Schwarz, "Improved bgp convergence via ghost flushing," in *Proc. of IEEE INFOCOM*, 2003.

[7] D. Pei, M. Azuma, N. Nguyen, J. Chen, D. Massey, and L. Zhang, "BGP-RCN: Improving BGP Convergence Through Root Cause Notification," UCLA CSD, Tech. Rep. TR-030047, October 2003, http://www.cs.ucla.edu/ peidan/bgp-rcn-tr.pdf.

[8] D. Pei, L. Wang, D. Massey, S. F. Wu, and L. Zhang, "A study of packet delivery performance during routing convergence," in *The International Conference on Dependable Systems and Networks (DSN)*, 2003.

[9] C. Labovitz, A. Ahuja, R. Wattenhofer, and S. Venkatachary, "The impact of internet policy and topology on delayed routing convergence," in *Proc. of IEEE INFOCOM*, 2001.

[10] D. Pei, , B. Zhang, D. Massey, and L. Zhang, "An analysis of path-vector routing protocol convergence algorithms," UCLA CSD, Tech. Rep. TR-040009, 2003.

[11] SSF Research Network, "Ssfnet," http://www.ssfnet.org.

[12] BJ Premore, "Multi-as topologies from bgp routing tables," http://www.ssfnet.org/Exchange/gallery/asgraph/index.html.

[13] D. Pei, , X. Zhao, D. Massey, and L. Zhang, "A study of bgp path vector route looping behavior," in *International Conference on Distributed Computing Systems*, Mar. 2004.

[14] L. Gao, "On inferring autonomous system relationships in the internet," *ACM/IEEE Transactions on Networking*, vol. 9, no. 6, pp. 733–745, 2001.

[15] Z. M. Mao, R. Bush, T. Griffin, and M. Roughan, "Bgp beacons," in *ACM SIGCOMM Internet Measurement Conference (IMC)*, 2003.

[16] J. Chandrashekar, Z. Duan, Z.-L. Zhang, and J. Krasky, "Limiting path exploration in path vector protocols," University of Minnesota, Tech. Rep., 2003.

[17] J. Luo, J. Xie, R. Hao, and X. Li, "An Approach to Accelerate Convergence for Path Vector Protocol," in *Proceedings of IEEE Globecom*, Nov. 2002.

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