A Technique for Reducing BGP Update Announcements through Path Exploration Damping

Geoff Huston, Mattia Rossi, and Grenville Armitage

***Abstract*—This paper defines and evaluates Path Exploration Damping (PED) – a router-level mechanism for reducing the volume of propagation of likely transient update messages within a BGP network and decreasing average time to restore reachability compared to current BGP Update damping practices. PED selectively delays and suppresses the propagation of BGP updates that either lengthen an existing AS Path or vary an existing AS Path without shortening its length. We show how PED impacts on convergence time compared to currently deployed mechanisms like Route Flap Damping (RFD), Minimum Route Advertisement Interval (MRAI) and Withdrawal Rate Limiting (WRATE). We replay Internet BGP update traffic captured at two Autonomous Systems to observe that a PED-enabled BGP speaker can reduce the total number of BGP announcements by up to 32% and reduce Path Exploration by 77% compared to conventional use of MRAI. We also describe how PED can be incrementally deployed in the Internet, as it interacts well with prevailing MRAI deployment, and enables restoration of reachability more quickly than MRAI.**

***Index Terms*—Border Gateway Protocol (BGP), Internetworking, Minimum Route Advertisement Interval (MRAI), Path Exploration Damping, Route Flap Damping (RFD), Routing**

I. Introduction

THE INTERNET'S inter-domain routing protocol, the Border Gateway Protocol (BGP) has been progressively refined since originally described in 1989 [1]. BGP is currently specified in RFC 4271 [2]. During BGP's lifetime the Internet has grown from less than 20,000 distinct routing entries in 1993 more than 300,000 routing entries in 2009 [3]. The underlying issues with routing scalability related to such an increase in routing domains (finer granularity of information and denser interconnection), have been persistent study themes in the investigation of the behavior of BGP since it's inception [4]. Early observations noted that BGP is a "chatty" protocol, and that the amount of transient and superfluous update messages sent by BGP appeared to contribute to unnecessarily high processing loads being imposed on BGP speakers, and potentially extended times for the routing system to converge to a stable state.

One modification intended to reduce the number of transient BGP updates was through the use of a Minimum Route Advertisement Interval (MRAI), described in RFC 1771 [5]. This mechanism allows a BGP speaker to announce routes about a certain destination (a prefix) to its peers no more frequently than once per MRAI time interval. This MRAI timer was intended to reduce the BGP Update load, at a cost of the increase in the time for a routing system to converge to a stable state [5].

It has also been observed that a relatively small number of advertised destinations are the cause of a large number of routing updates by repeatedly announcing, withdrawing and then re-announcing their prefix (often at high frequency, and over extended periods of time) [6]. In order to suppress such pathological routing behavior in the network, Route Flap Damping (RFD) was added to BGP in 1993 and standardized in 1998 [7]. Related research and operational analysis however have discounted RFD efficacy [8], [9] and current operational advice is not to deploy this behavior [10]. While the original description of the MRAI timer applied to announcements, RFC 4271 described the application of the same MRAI timer to withdrawals. This mechanism, Withdrawal Rate Limiting (WRATE), has been observed to have a negative effect on average convergence times in BGP [11].

This paper proposes the use of Path Exploration Damping (PED) as a more effective alternative to RFD, MRAI, and WRATE for suppressing BGP's protocol behavior when BGP undertakes certain forms of "path exploration" [6], [12], [13]. Like RFD, PED suppresses the announcement of certain routes, but PED uses a timer that is related to MRAI behavior, and uses an update suppression selection algorithm that is related to the behavior of the AS PATH attribute of the route across successive updates. All other updates and all withdrawals are propagated to a BGP speaker's peers without further delay. We show that PED reduces the volume of BGP Update messages compared to existing damping mechanisms, without generating undue additional overhead for the BGP speaker.

A 2007 Internet draft identified two stages of BGP convergence – *reachability* and *optimality* – and classified reachability as the more important state to reach quickly [9]. We analyze the impact of RFD, WRATE, MRAI and PED on both stages and show that, although PED sometimes achieves optimality slower than MRAI, it always achieves reachability as fast as, and sometimes faster than, MRAI.

Sections II and III of this paper recap the underlying BGP dynamics and discuss related work. Section IV describes PED itself, while Section V outlines our reference implementation and evaluation methodology. Section VI presents our analysis of the real datasets from AS 131072 (Asia Pacific Network Information Center, APNIC) and AS 6447 (University of Oregon Route Views Project, Routeviews), which we use to show the impact of PED and its potential for incremental deployment in Section VII. In Section VIII we evaluate the effect of PED on BGP convergence, and conclude in Section IX.

Manuscript received 11 August 2009; revised 5 January 2010.

Geoff Huston is with APNIC, Canberra, Australia (e-mail: gih@apnic.net).

Mattia Rossi and Grenville Armitage are with the Centre for Advanced Internet Architectures, Swinburne University of Technology, Melbourne, Australia (e-mail: {mrossi,garmitage}@swin.edu.au).

Digital Object Identifier 10.1109/JSAC.2010.101005.

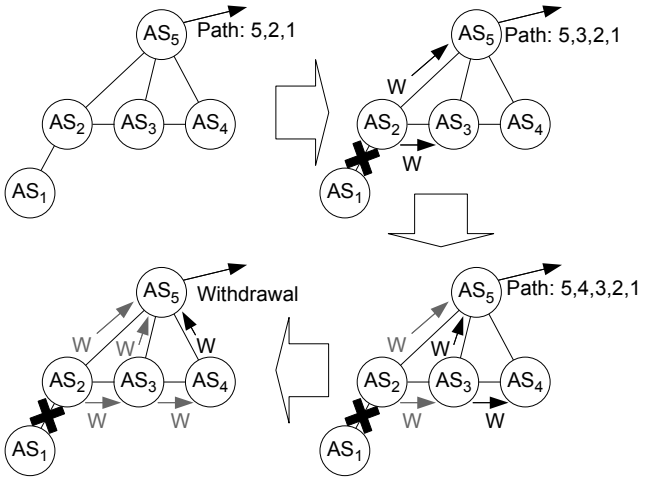


Fig. 1. Path Exploration triggered by one withdrawal

TABLE I
BGP UPDATE TYPES RELEVANT TO PATH EXPLORATION

Code	Description
AA+	Announcement of an already announced prefix with a longer AS Path (update to longer path)
AA-	Announcement of an announced prefix with a shorter AS Path (update to shorter path)
AA0	Announcement of an announced prefix with a different path of the same length (update to a different AS Path of same length)
AA*	Announcement of an announced prefix with the same path but different attributes (update of attributes)
AA	Announcement of an announced prefix with no change in path or attributes (possible BGP error or data collection error)
AW	Withdrawal of an announced prefix
NA	Announcement of a previously unknown or withdrawn prefix

II. BACKGROUND

A. Basic BGP Dynamics

Two types of events trigger BGP update messages: reachability failure and reachability discovery.

In the case of reachability failure, the notifying Autonomous System (AS) will generate a BGP withdrawal that is intended to propagate through the network. This propagation of a withdrawal may trigger consequent announcements, as is the case when a BGP speaker believes that it has a choice to switch to a less preferred, but still valid path upon receipt of the withdrawal. In this case the BGP speaker involved does not propagate the withdrawal explicitly, but will propagate an announcement for the remaining valid path to its BGP peers[1], which is appropriately interpreted as an implicit withdrawal of its previous announcement and an announcement of a new best path. Further announcements may be triggered as the routing change to this backup path propagates across the network. In some cases this appearance of an apparent backup path is a transient artifact of differing propagation times of withdrawals across different routing paths in the network. In such cases the announcement of the backup path is a short-lived announcement, possibly followed by further successive announcements of alternate paths, and ultimately by a withdrawal once the original withdrawal has been propagated through all possible routing paths. This routing protocol behavior is known as Path Exploration[2] [12], [13].

Cyclic failures, where an inter-domain connection fluctuates between being operationally active and out of service at a high frequency, can generate a high volume of updates. Depending on the density of interconnections in the BGP network, the number of such updates can multiply rapidly [6].

A simplified example of how path exploration is induced is depicted in Figure 1. When the link between ASes 1 and 2 (AS_1 and AS_2) fails, AS_2 sends withdrawal W to AS_3 and AS_5. AS_5 temporarily believes AS_1 is still reachable via AS_3, and advertises path $\{5,3,2,1\}$. AS_3 forwards W to AS_4 and AS_5, leading AS_5 to now believe (and advertise) path $\{5,4,3,2,1\}$ as valid. Finally, AS_4 forwards W to AS_5, who withdraws the path. During this time, AS_5 has announced to the world a rapid sequence of successively longer paths involving AS_1.

Using the categories of update types in Table I – an extract of update type classification used in [14] – AS_5 can be said to emit an update sequence {NA, AA+, AA+, AW}, with the unnecessary {AA+, AA+} indicative of Path Exploration. The system is converged after the final AW.

B. Minimum Route Advertisement Interval

The introduction of MRAI in BGP was intended as a simple solution to suppress the multiplication of such Path Exploration messages. Use of MRAI essentially imposes a low-pass filter on updates for a given address prefix, limiting the frequency of announced changes to a maximum of one per MRAI interval. Once an update has been sent by a BGP speaker to its BGP peers for a given prefix all further updates for that prefix are to be suppressed until the expiration of the MRAI timer, at which time only the current prefix state is announced to the BGP speaker's peers. Any intervening transient routing states are not announced to the BGP Speaker's peers, as any updates for that prefix that were queued during this MRAI interval, except the final update, are suppressed.

While MRAI has been deployed widely through the Internet, the common implementation of the MRAI behavior is subtly different from the description above. Instead of using a per-prefix update suppression timer, the generally available implementations of MRAI in BGP use a per-peer announcement timer. All updates to a BGP peer are queued by the BGP speaker. Successive updates to the same prefix cause the previously queued updates for that prefix to be flushed from the output queue, as a form of "queue compression". At the expiration of the MRAI timer the entire output queue state is sent to the BGP peer, and the queue is flushed. The timer is restarted and the queue is reopened for another MRAI interval.

C. Withdrawal Rate Limiting and Route Flap Damping

Situations of persistent short term transient reachability failure have motivated the adoption of RFD and WRATE. WRATE is the application of MRAI to withdrawals, effecting

[1]We refer here to eBGP peers over inter-AS connections, disregarding iBGP (interior BGP) peers on intra-AS connections which are subject to slightly different behavior

[2]Or sometimes also referred to as Path Hunting [9]

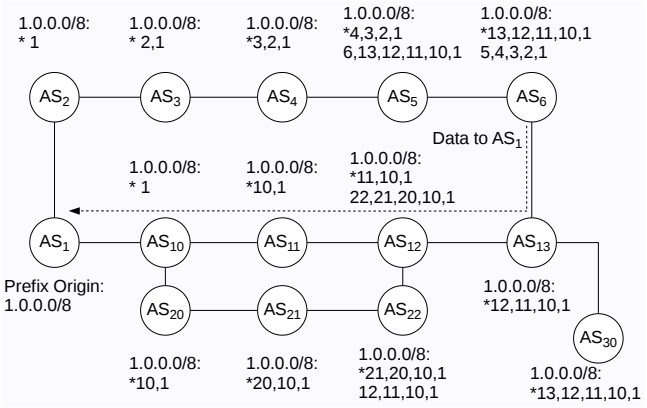


Fig. 2. An example converged system. Data originating from AS_6 is routed to AS_1 via AS_{12},AS_{11} and AS_{10}. Each AS is tagged with the prefix and path(s) through which the prefix is reachable. Preferred path marked with "*".

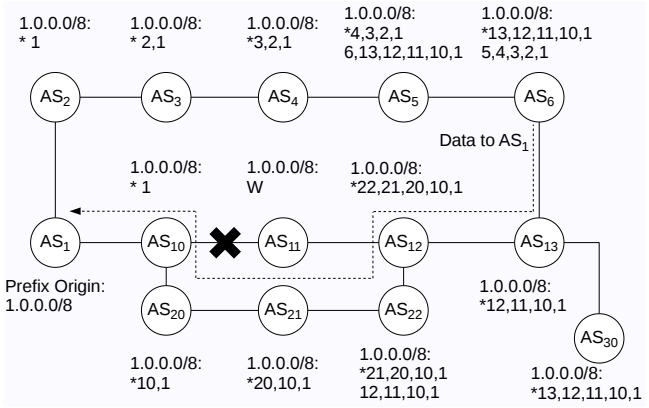


Fig. 3. If the link between AS_{10} and AS_{11} fails, reachability is already restored when AS_{12} gets the update

a low-pass filter on withdrawal propagation. As with MRAI for updates, WRATE aims to allow a BGP speaker's peers some time to converge to a stable state before the local BGP speaker propagates the prefix update to its peers.

RFD is an entirely different mechanism, which uses a prefix's past routing instability characteristics from a given BGP peer to determine if the BGP speaker should continue include the peer's advertisement for the prefix in the current local best path selection algorithm. Each update and withdrawal for a prefix from a BGP peer adds to the local "instability penalty score" for the peer and prefix pairing, while the score will decay over time when there is no further update activity. Whenever this score exceeds a threshold value the peer's announcement is ignored by the local route selection process. A prefix that is unstable at source would generate a history of instability across the entire inter-domain routing system, and BGP speakers would react by progressive damping of the unstable prefix via the RFD mechanism. An unstable transit link would cause a similar damping response, but in this case any alternate stable paths that did not use the link in question would continue to be announced, with the result that the unstable link would be withdrawn from the routing system for an extended period. The RFD specification suggests a suppression interval of between 30 and 60 minutes [7].

RFD has a negative impact on average BGP convergence times, as many of the behaviors that trigger an RFD response were observed to be actually caused by the Path Exploration behavior of the protocol rather than by any underlying physical instability in the network's infrastructure. Consequently, short term protocol behaviors at the one second level of granularity were causing RFD-generated reachability failure events on a scale of hours – RFD's response was disproportionate to the nature of the triggering event [11], [8].

D. BGP Convergence: Optimality vs. Reachability

As previously noted, convergence in routing can be divided into two distinct parts: optimality and reachability [9].

Optimality ensures that *every* BGP speaker knows *the best* route to the destination of a packet. Reachability ensures that there is always *a possible* route to the destination – even if the route is suboptimal. Although optimality and reachability are sometimes achieved at the same time, there are certain situations in BGP, where reachability is restored long before optimality is achieved. Furthermore, e.g. during a path exploration event, convergence in terms of reachability might be achieved many times, while convergence in terms of optimality is not achieved at all.

This is most often the case on T_{long} and T_{short} events on reachability failure. A T_{long} event is where a failure affects only a portion of the possible paths to a destination like a single link failure of a multihomed AS. A T_{short} event is the recovery after such a failure, and a better (usually shorter) path to the destination becomes available [6], [15]. These events cause legitimate AA0, AA- and AA+ updates, which can eventually also transform into withdrawals (AW). With the right mix of delay and topology, T_{long} and T_{short} might also generate path exploration.

Reachability and optimality overlap instead in the case of T_{down} and T_{up} events. A T_{down} event is a failure that affects all paths to the destination (the destination becomes completely unreachable), and a T_{up} event marks a recovery from such a failure or the announcement of a previously unknown prefix (a new announcement) [6], [15]. These events should only originate AW and AN updates throughout the system, but can also generate path exploration events.

Convergence on T_{down} and T_{up} events can be explained as follows: the T_{down} has to be notified through the network as quickly as possible, to avoid unnecessary traffic load on links – rapid achievement of optimality is desirable, reachability is not relevant in a T_{down} event. The T_{up} event has to be notified quickly throughout the whole network as well, as reachability can only be achieved when every BGP speaker knows about the availability of a path to the destination. Optimality and reachability overlap in this case.

Upon T_{long} and T_{short} events, reachability and optimality are achieved at two completely different times. On a T_{long} event, reachability is restored much quicker than optimality as depicted in Figures 2 and 3: Once AS_{12} receives the withdrawal from AS_{11} after the link between AS_{10} and AS_{11} fails (the T_{long} event), it installs the route via AS_{22}. Reachability (of AS_1) is restored immediately for every BGP speaker except AS_{11}. One more update is needed and AS_{11} can reach AS_1 as well.

We can say that reachability is already restored for every BGP speaker further upstream of the *first* BGP speaker on any

given route that has a valid alternative route to the destination (although less preferred, and usually with a longer AS path), when this BGP speaker (which we call *altBGP*) has converged. Reachability is further restored for every BGP speaker further downstream of *altBGP*, when such a BGP speaker receives the new route advertised by *altBGP*.

On the other hand, optimality might not be reached for a long period of time. As shown in Figure 4, AS_6 will chose the new best path only two updates later. The optimal path at AS_{13} will be installed even later. Within this instability period, AS_1 has been reachable all the time.

On a T_{short} event, reachability is irrelevant as it is already achieved. It only matters to achieve optimality, and thus to propagate the updates regarding the event (the new best path) as quickly as possible to the ASes previously affected by the T_{long} event.

E. Impact of MRAI, WRATE and RFD on Convergence and Path Exploration

As WRATE is delaying withdrawals in the BGP system, it has a negative impact on convergence. Legitimate withdrawals on T_{down} events will be delayed, putting unnecessary load on links further upstream (in the form of traffic that otherwise would be dropped earlier). On T_{long} events, it delays updates to *altBGP*, affecting reachability! Although WRATE has been introduced to reduce the number of updates generated by path exploration and route flapping, it is limited by an inability to distinguish such updates from legitimate updates. Like WRATE, RFD is also unable to differentiate between the type of updates. It is also possible that RFD suppresses withdrawals which are legitimate, creating a similar effect to WRATE on convergence. The delaying of withdrawals, and the mixture in the BGP routing system of BGP speakers that do and do not use these mechanisms, exacerbates path exploration.

On the other hand, MRAI does not delay withdrawals. This ensures reachability and optimality in case of a T_{down} or T_{long} event. However, in the case of a T_{up} or T_{short} event, MRAI affects convergence time because updates are delayed. The specific convergence delay depends a lot on the implementation of MRAI. If the MRAI timer is implemented on a per-peer basis some peers might converge slower than others, due to the interaction of independent MRAI timers used by each BGP speaker. If the MRAI timer is implemented on a per-peer/per-prefix basis, the delay will be mostly the same for all peers. In addition, implementations may use techniques like Path-Based Poison Reverse, where the BGP speaker explicitly sends a withdrawal to those peers who are listed in the AS Path of the currently selected "best" route, and improve convergence time.

III. Related work

In a series of experimental analyses of BGP, Labovitz et al. observed that variations in MRAI timers across a BGP network, the deployment of RFD, highly interconnected ASes and the need for multihoming ASes exacerbate the amount of Path Exploration events, and that Path Exploration is a major cause of delayed BGP convergence [6], [12], [13], [16], [17]. A number of approaches have been proposed to reduce the update load associated with Path Exploration and improve overall convergence time.

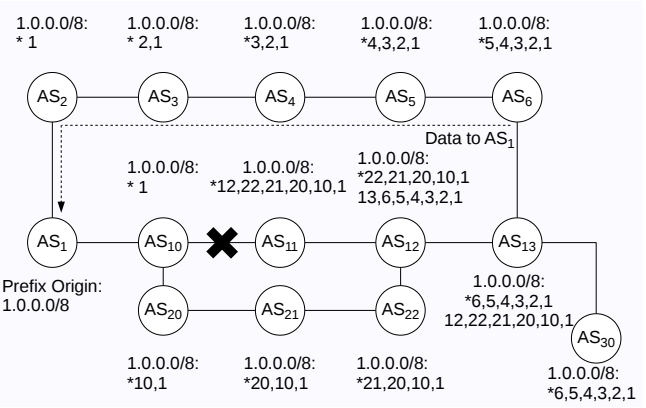


Fig. 4. Control plane convergence is achieved later than data plane convergence. Reachability is ensured during the whole control plane instability period

A. Sender-Side Loop Detection

Labovitz et al. in 2000, have also proposed the use of Sender-Side Loop Detection (SSLD), a method to decrease convergence time in the case of reachability failure [6]. SSLD is a variant of BGP where the sender does not propagate an announcement to an eBGP peer when that peer's AS number is already in the AS path of the update: The sender detects the loop and does not propagate the update, rather than conventional BGP, where the receiver has that role. Using this technique, convergence can be reached faster, and the risk of generating Path Exploration events is diminished. As SSLD is incompatible with the current transition strategy for deploying 4-byte AS numbers, it is unlikely to be actively deployed in the Internet[3].

B. Consistency Assertions

In 2002 Pei et al. suggested to improve BGP convergence by consistency assertion on route announcement and withdrawal [18]. After every received update message, routes contained in the RIB are tested for conflict with the newly arrived information, and the result is taken into consideration in the path selection process. This approach has been shown to diminish the volume of superfluous update messages, and improve average convergence time. The additional load imposed by consistency assertions would have a negative impact on CPU and memory usage and update processing time.

C. Ghost Flushing

In 2003 Bremler-Barr et al. proposed a method called "Ghost Flushing" [19] to alleviate the instability and convergence time problem. They eliminate Path Exploration events from the network by sending an explicit withdrawal of the prefix to all peers if a switchover to a less preferred route happens. It is unclear what impact this approach would have

[3]4-byte ASes represent themselves as AS23456 to 2-byte ASes. Thus a BGP speaker using 2-byte AS numbers, sitting between two BGP speakers enabled for 4-byte AS numbers, would 'see' AS 23456 in both upstream and downstream directions, erroneously triggering SSLD.

if incrementally deployed in the current Internet. (In the presence of extensive use of the MRAI timer and RFD, Ghost Flushing would generate typical flapping behavior, by preceding announcements by withdrawals.)

D. Root Cause Notification

In 2004 Pei et al. proposed "Root Cause Notification" (RCN) [20], following the idea of consistency assertions, by allowing a BGP speaker to detect and ignore invalid announcements immediately. RCN tries to offload the consistency assertion from the routers to the network, by adding a BGP option which carries information of the root cause of a link failure. With this information, all routes affected by the failure can be easily deleted from the RIB and invalid announcements can be avoided. RCN would add minimal overhead to routers, but it changes the BGP protocol over-the-wire.

E. Differentiated Update Processing

In 2006 Sun et al. introduced "Differentiated Update Processing" (DUP) [21], claiming it reduces BGP updates by 30% and improves convergence time by 80% by putting BGP updates in different classes and sending announcements depending on the class they are in. The BGP update classification depends on the novelty of an update and on the traversed topology. The idea can be incrementally deployed, and does not require any changes to the BGP protocol over-the-wire. The authors also claim that DUP creates little overhead on CPU and memory usage.

IV. Introducing Path Exploration Damping

In 2007 Huston proposed Path Exploration Damping[4] (PED) as an easy-to-deploy alternative to MRAI and RFD, focusing on the Path Exploration updates leading to transitory routing states known to delay BGP convergence time [14].

A. Defining Path Exploration Damping

PED aims for easy implementation and to be effective if deployed in the current MRAI-dominated Internet. It uses a similar approach for improving BGP convergence time as the methods described in Section III, it does not alter the BGP protocol over-the-wire, and introduces little additional CPU and memory overhead on routers.

Algorithm 1 explains PED in detail. Referring to the update message classification in Table I, PED delays update messages which would announce a route with a same-length or longer AS Path than the previously announced route for the same prefix (AA+, AA, AA* or AA0 updates) for a period of time we call the "Path Exploration Damping Interval" (PEDI). As per BGP protocol only the best path gets announced, AS Path length comparison takes place between the previously announced best path and the newly selected best path.

Shorter path announcements (AA-, AW and NA update messages) are not delayed.

PED will also impact on non-Path Exploration sequences of {AA+, AA+, etc} that do not end with AW or AA-. In such cases the final update will be emitted up to one PEDI later than it would have been without PED. Unlike MRAI, we extend the damping period by the PEDI each time a candidate update is detected.

Algorithm 1 Path Exploration Damping

On a per-peer, per-prefix basis define:

- a new *path exploration damping interval* (PEDI) Timer
- a temporary outbound queue for holding an update

When ready to transmit an update for known prefix X:

1) If sending an AA+, AA, AA* or AA0 update and
   - the PEDI Timer is not active: Queue the update, and start the PEDI Timer
   - the PEDI Timer is active: Delete any previously queued update for this prefix, queue the new update and restart the PEDI Timer from zero
2) If sending an AW or AA- update:
   - Eliminate any previously queued update for this prefix and send the new update immediately

An NA update (involving a previously unknown or withdrawn prefix) is transmitted without damping

B. Impact of PED on convergence and Path Exploration

PED is based upon ensuring reachability in a BGP system first, and achieving optimality as a secondary goal. In cases where reachability doesn't matter or overlaps with optimality, PED aims to achieve optimality.

Referring to section II-D, this is the case on T_{down}, T_{up} and T_{short} events. A T_{down} event only generates withdrawals, which are not suppressed, and propagated immediately through the whole routing system. If withdrawals "run late" at an *altBGP* speaker, the alternative path, which per the BGP protocol has to be longer[5], the risk is that an AA+ update is generated and shortly after the "late" AW update follows – the T_{down} event is interpreted as T_{long}. This is the most typical way Path Exploration events are created in the Internet. In any case, PED will delay the AA+ update for enough time to allow the "late" withdrawal to arrive. The withdrawal then generated will be propagated immediately after – and possibly even before – the PEDI timer expires. This way the generation of updates by Path Exploration is suppressed.

Following BGP logic, T_{up} events will always generate NA type updates. NA updates are processed without delay, allowing a route to be propagated as quick as possible. T_{short} events will always generate NA type updates to the *altBGP* speaker, and from there change into AA- or AA0 type updates. Again optimality is achieved quickly, as NA and AA- updates are not delayed. In the case of AA0 updates, it is difficult to decide whether one of two paths with the same length is preferable to the other, so PED treats them like AA+ updates. If an NA or AA- changes into an AA0 type update, it means the BGP speaker that changed the update already had an alternative path, so ensuring reachability.

[4]Initially called "Update Damping"

[5]The AS Path length comparison is generally the relevant part in the BGP decision process

On the other hand, T_{long} events generate withdrawals at the origin, which can mutate into AA+ events at an *altBGP* speaker. The important part here is to not delay withdrawals on their way to *altBGP*, so that *altBGP* may switch to the alternative path to ensure reachability. In addition, AA+ messages are sent immediately from *altBGP* downstream to the BGP speakers that delivered the withdrawals, as these updates are interpreted as NA. Optimality is delayed in this case for all BGP speakers upstream of *altBGP*, as AA+ updates are delayed on each BGP hop. As with T_{down} events, in this case delaying the announcement of suboptimal routes allows us to suppress possible generation of Path Exploration events caused by delayed withdrawals.

PED drops queued announcements which are superseded by a new announcement or a withdrawal – although AW and AA- updates are forwarded immediately, AA+, AA, AA* or AA0 updates effectively experience output queue compression. This behavior can negatively impact optimality. However, it does not interfere with reachability – the BGP speaker still has a valid forwarding path toward the affected prefix.

C. Interaction of PED, routing policies and BGP decision process

By selecting the routes to delay based on the AS path length attribute, PED minimizes CPU and memory overhead. The AS path length is computed for every prefix and stored every time the BGP decision process is executed, and PED is applied after this decision process.

In certain cases, using the AS path length for comparison might interact negatively with routing policies deployed on a BGP speaker. The decision process of a router might prefer one route over another depending on the LOCAL_PREF attribute set by routing policies, rather than the AS Path length, thus preferring a an AA+, AA0, AA or AA* update over an AA- update. In such a case, PED would delay the announcements for a PEDI interval.

As the BGP speaker will install the route selected by LOCAL_PREF in the RIB before PED is applied, such a BGP speaker acts like an *altBGP* speaker, ensuring that the packets are always forwarded along the correct path, while upstream routers get the information with a delay – reachability is ensured, optimality delayed.

(It would be possible to modify PED in case a route is selected by LOCAL_PREF, thus never delaying such announcements. But it has to be kept in mind that a PED enabled BGP speaker further upstream might delay that update anyway.)

D. PED and flapping routes

Flapping routes manifest by constantly withdrawing and announcing a route ({AW NA}* sequences) or more likely by repeatedly switching paths (mixed sequences of AA+, AA, AA*, AA0 or AA-) within a short time frame. In the first case, PED would impact beneficially if deployed at an *altBGP* speaker, where it would transform the sequence into an AA- {AA}* sequence. If the route would flap quicker than the PEDI interval, the AA update would be delayed until the route has stabilized. The *altBGP* speaker ensures reachability all the time. In the second case, if the BGP speaker generating a flapping route is a peer of a PED enabled BGP speaker, PED would immediately transform the update sequence into an AA- {AA}* sequence. The route to the flapping peer would be always valid in this case, thus ensuring reachability all the time.

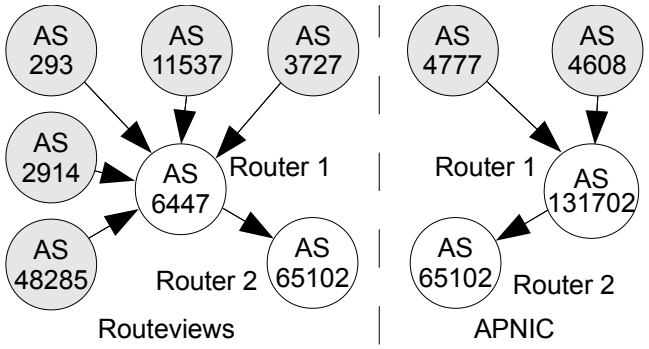


Fig. 5. The BGP network recreated using APNIC and Routeviews data

V. Experimental Methodology

We have implemented PED (as described in Algorithm 1) in Quagga version 0.99.13 [22], [23]. By implementing PED in Quagga's output queue (the Adj-RIB-Out in RFC 4271's abstract model [2]) the PEDI Timer operates on a per-peer and per-prefix basis[6].

BGP implementations typically apply jitter to the MRAI Timer. We applied a similar concept to the PEDI Timer, subtracting a random value between 0 and 3 from the initial PEDI value each time the PEDI Timer is reset.

The first part of our experimental analysis of PED utilized accelerated playback [24] of inbound IPv4 BGP updates previously collected at AS 131072 (APNIC) and AS 6447 (Routeviews). Figure 5 illustrates our two scenarios – a private AS 65102 (Router 2) collecting BGP updates[7] from either AS 131072 or AS 6447 (Router 1, configured to emulate AS 6447 or AS 131072 as required). Both routers were actual instances of Quagga with our PED extensions.

Using real BGP update data generated by BGP speakers in the Internet eliminates the possibility of exactly classifying the nature of the event that triggered an update collected at one of our collection points[8]. Nevertheless, we can approach such a classification by observing the update sequences per prefix found in such data. Update sequences are consecutive updates about a prefix until the prefix is converged. Convergence itself is not easily detectable without the knowledge of the originating event – we must guess the end of an update sequence by determining a "stability interval", a period of time applied on a per prefix and per peer basis, for which no update for this prefix has arrived from the same peer.

If the last update of such an update sequence is a withdrawal (AW), the event that caused the update sequence has either to

[6]Although [14] initially envisaged PED being applied to messages in the routing table, applying it in Quagga's output queue also enables support for output queue compression.

[7]BGP updates collected at arrival in Quagga's input queue on Router 2 (Adj-RIB-In of the abstract BGP model)

[8]Contrary to a "BGP update message" which possibly contains a mixture of announcements and withdrawals as sent over the wire, in this case we use *update* to mean either an announcement or a withdrawal

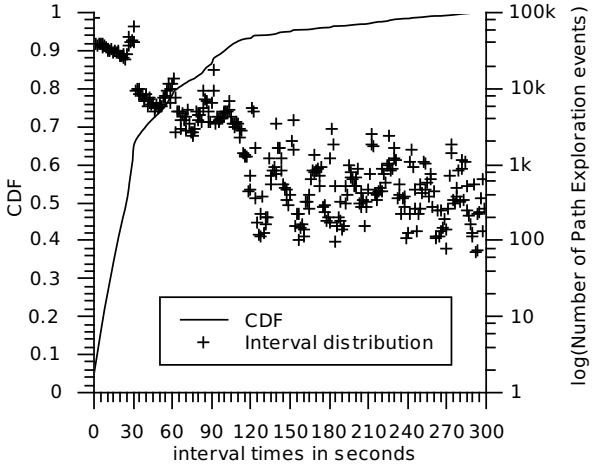


Fig. 6. Distribution of Path Exploration event intervals seen at the input side by AS 131072 (mid April to mid May 2009)

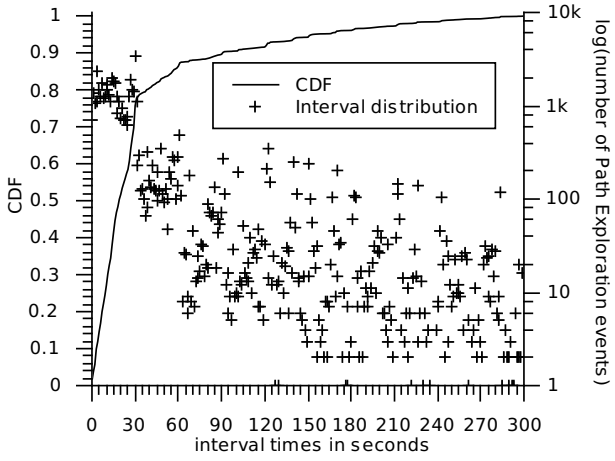


Fig. 7. APNIC: Distribution of Path Exploration intervals over 24hrs - input side

be a T_{down} or a T_{long} event[9]. In any case, if the update sequence is longer than the single withdrawal, we have detected a Path Exploration event.

Where a sequence ends in an NA or AA-, the event has either been a T_{up} or T_{short} event, as can be deduced by the convergence and Path Exploration analysis in section II-D and IV-B. We call this further convergence to a shorter path. If an update sequence ends in an AA+ we can be sure that it is generated by a T_{long} event. An ending in AA0, AA* or AA can be caused by a variety of situations, as explained in section IV-B. In any case we say the sequence converges to a longer (or equal-length) path.

The update sequence analysis allows us to quantify the number of Path Exploration events experienced at a single BGP speaker, without needing to know what event caused it. However, knowing the duration of an update sequence and the number of updates it consists of does not allow us to determine convergence time. Convergence time is considered the time it takes for an update to arrive from the generator of the update to the BGP speaker being observed.

We have analyzed convergence time with a simulation, based upon the topology described in Figure 2, using our modified version of Quagga.

The following sections illustrate the impact of PED on:

- The total number of announcements, withdrawals and BGP update messages sent, compared to using MRAI or no output queuing at all[10]
- The number of Path Exploration events compared to MRAI or no output queuing
- Piecemeal deployment – interoperability of MRAI and PED
- Convergence time, compared to using MRAI

VI. Path Exploration events and update sequences observed in APNIC and Routeviews datasets

In this section we summarize some relevant characteristics of the Routeviews and APNIC datasets. Preliminary analysis shows that Path Exploration is still quite prevalent, and that observed update sequences can appear quite different depending on the location of one's collection points. We evaluate update sequences using a range of stability intervals (as defined in section V) and chose a stability interval of 300 seconds for later evaluation of PED's impact on convergence.

A. Path Exploration events over a month and 24 hours

Figure 6 shows the distribution of intervals between updates during Path Exploration events[11] received over a month (mid April to mid May 2009) by AS 131072 from two external peers (AS 4777 and AS 4608). Path Exploration events are detected on a per-peer basis. Roughly 80% of all Path Exploration events are less than 60 seconds apart, while 60% are less than 30 seconds apart.

Distinct peaks around 30 and 60 seconds (and higher multiples of 30) are due to upstream peers using a default 30 second MRAI Timer[12].

In order to simplify subsequent analysis, we extracted 24 hours of BGP updates (April 30th to May 1st 2009) from the APNIC and Routeviews datasets. Our 24 hour periods were representative of the month from which they were extracted. Figure 7 shows the APNIC 24 hour data's distribution of Path Exploration events, which is broadly similar to that of Figure 6. Figure 8 shows the Routeviews 24 hour data's distribution (similar to Routeview's monthly distribution, which is omitted to save space).

B. Update sequence analysis over 24 hours

Figure 9 shows the total number of BGP update messages, prefix announcements and prefix withdrawals over the 24 hour period for the input datasets of AS 131702 and AS 6447.

Going into more detail, the data is composed by update sequences (Figures 10, 11, 12) and single announcements and withdrawals (Figures 13, 14). Even if the composition and number of update sequences and the number single

[9]In the case of multiple events affecting a prefix, it is only possible to detect the most recent

[10]We used MRAI Timer values of 30 and 0 sec, values commonly recommended by major router vendors

[11]Intervals between successive updates in a {{AA+}*, AW} sequence.

[12]Updates for the same prefix arriving from both AS 4777 and AS 4608 also result in intervals shorter than 30 seconds, because of jitter and the upstream 30-second MRAI Timers being unsynchronized.

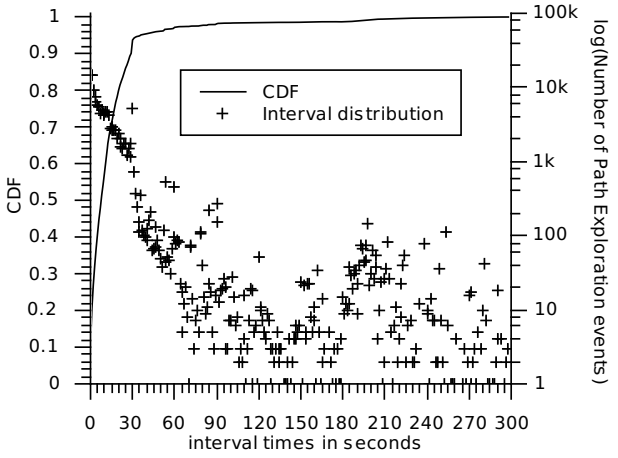


Fig. 8. Routeviews: Distribution of Path Exploration intervals over 24hrs - input side

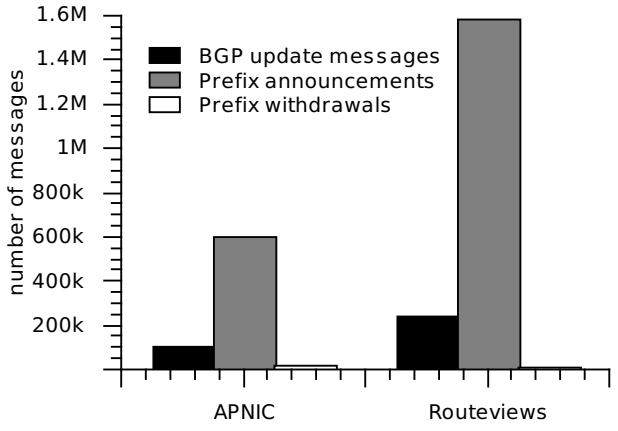


Fig. 9. APNIC and Routeviews input data: Number of BGP update messages, prefix announcements and prefix withdrawals

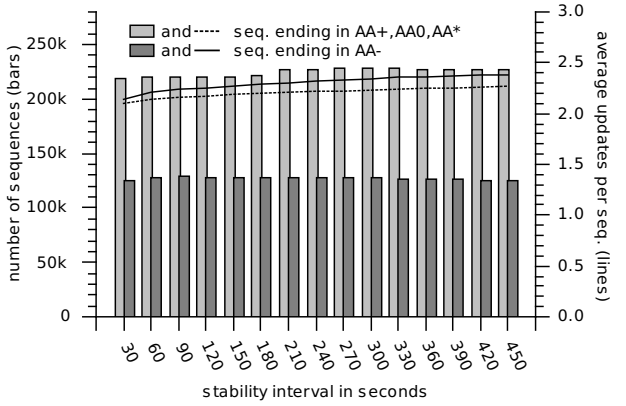


Fig. 10. Analysis of update sequences ending in longer and shorter paths using the APNIC dataset

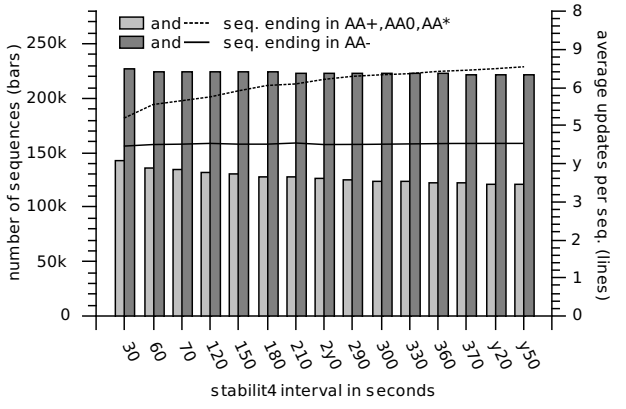


Fig. 11. Analysis of update sequences ending in longer and shorter paths using the Routeviews dataset

announcements or withdrawals varies with each choice of stability interval, the total number of updates as per Figure 9 remains consistent.

The update sequences are grouped into sequences ending in an announcement or a withdrawal, where we distinguish between announcements of a longer (or equal-length) path and a shorter path. Sequences ending in a withdrawal are Path Exploration events.

Figure 10 reveals that at the APNIC collection point more update sequences end in longer (or equal-length) paths than shorter paths. The number of sequences, and number of updates per sequence, is relatively constant with increasing stability intervals.

Figure 11 reveals that at the Routeviews collection point more update sequences end in a shorter path than a longer (or equal-length) path. The number of update sequences ending in a longer or equal path drops off very slowly with increasing stability interval. The average number of updates per sequence seems to increase less rapidly for the same stability intervals. The number of update sequences ending in a shorter path seems relatively stable with increasing stability intervals.

The number of Path Exploration events seen in APNIC data peaks at a stability interval of 60 seconds. Routeviews data reveals broadly similar levels of Path Exploration events when using stability intervals of 30 or 60 seconds (due to the high number of updates seen in Figure 8 with intervals shorter than 30 seconds).

As we increase the stability interval, multiple Path Exploration events (originating at various points in the Internet) can result in a mixture of {{AA+}*, AW} sequences at the collector being treated as a single event. Consequently we see a drop off in the number of Path Exploration events and an increase in the average number of updates per event.

C. A suitable stability interval for analyzing PED's impact

To analyze PED's impact on Path Exploration events we identify the shortest stability interval that captures essentially all update sequences ending in a withdrawal. We do this by looking at how many update sequences consist of a single announcement, and in particular a withdrawal, as a function of stability interval. (If a stability interval is too short, we may accidentally count two or more announcements as multiple update sequences when they are actually part of a single update sequence.)

Figure 13 shows the number of update sequences consisting of a single announcement (of any type) within a given stability interval. More such update sequences are observed at APNIC than Routeviews.

Figure 14 shows the number of update sequences consisting of *single withdrawals* observed at AS 131072 and AS 6447. At stability intervals around 300 seconds this drops almost to zero for both APNIC and Routeviews datasets. In other words,

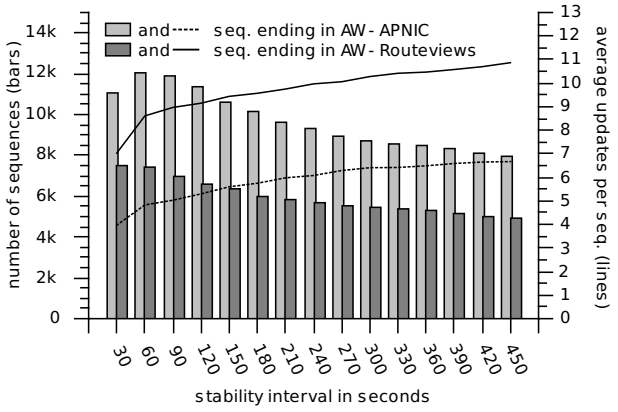


Fig. 12. Analysis of update sequences ending in a Withdrawal (Path Exploration events) using the APNIC and Routeviews dataset

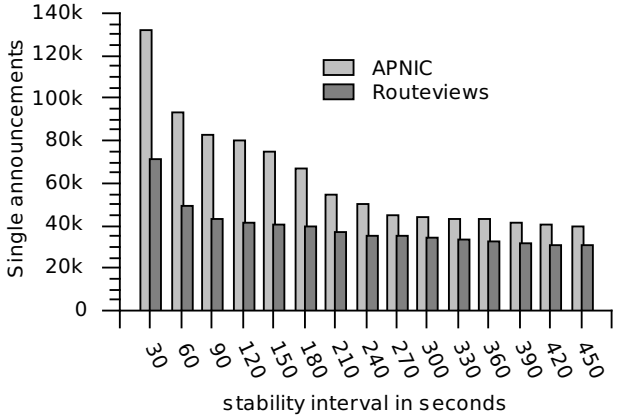


Fig. 13. APNIC and Routeviews update sequences consisting of single announcements

a stability interval of 300 seconds is sufficient to ensure we correctly identify essentially all Path Exploration events (and do not overcount late-arriving withdrawals as separate, single-announcement update sequences). We use this stability interval in the rest of this paper when analyzing the impact of PED, MRAI or no output queuing (MRAI with 0 seconds).

VII. Evaluation of PED with Real Data

In this section we first evaluate the *impact on Router 2* when Router 1 is fed with 24 hours of either the APNIC or Routeviews dataset, and utilizes either PED or MRAI on its output toward Router 2 (sub-section VII-A). Then we evaluates the possibility of incremental deployment of PED in the Internet (sub-section VII-B).

A. Using PED or MRAI at Router 1

We evaluate MRAI with MRAI Timers of both 30 and 0 seconds for both APNIC and Routeviews datasets. For the APNIC dataset we evaluate PED with PEDI values between 30 and 70 seconds in steps of 5. For the Routeviews dataset, due to significant level of Path Exploration intervals below 30 seconds[13], we evaluate PED using PEDI from 5 to 70 seconds in steps of 5.

[13]We suspect some of the peers were using 0 second MRAI

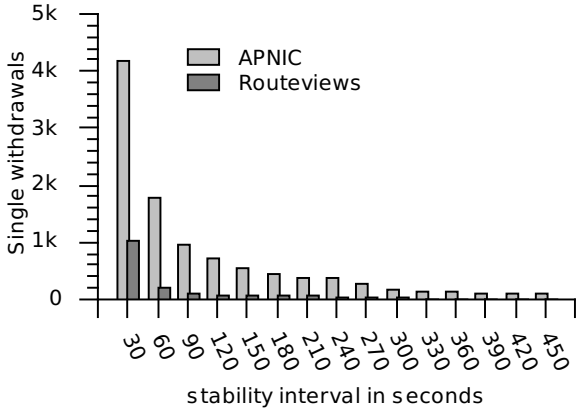


Fig. 14. APNIC and Routeviews update sequences consisting of single withdrawals

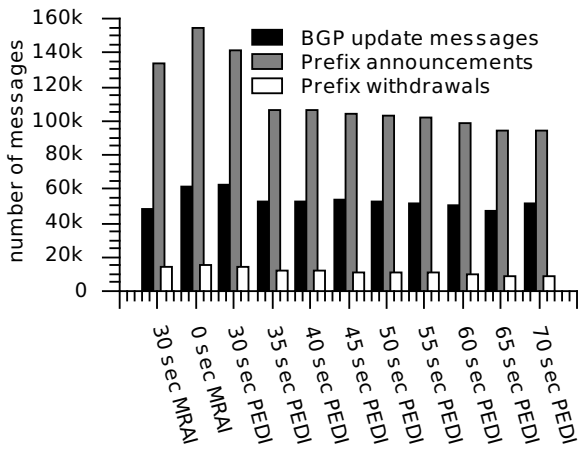


Fig. 15. APNIC scenario: Number of BGP update messages, prefix announcements and prefix withdrawals

1) Impact on total number of updates: First we evaluate the total number of prefix announcements and BGP update messages (the actual BGP message on the wire) generated by MRAI and PED over the 24h period for the APNIC and Routeviews dataset.

Figure 15 shows that for the APNIC dataset, PED with PEDI of 35 seconds or higher leads to a significant decrease in total prefix announcements relative to either MRAI scenario. From 134786 announcements using a standard MRAI of 30 seconds, PED with a PEDI of 35 seconds and 65 seconds produces 20% (107349) and 29% (95763) fewer announcements respectively (consistent with the predictions in [14]). However, the number of BGP update messages is slightly higher (within a 10% range) for PED with PEDI values of 35 seconds or more than for a standard MRAI of 30 seconds. (Using MRAI with 30 seconds allowed better packing of multiple prefix announcements into single BGP update messages.)

Figure 16 shows the total announcements, withdrawals and BGP update messages for the Routeviews dataset. Relative to an MRAI of 0 seconds, PED shows noticeable reductions in total announcements for all PEDI values from 5 to 70 seconds. (Even 5 and 10 second PEDI decrease the total announcements by 11527 (7%) and 24949 (15%) respectively.) Relative to an MRAI of 30 seconds, PEDI values of 25 seconds and greater show reductions in total announcements – 35 and 65

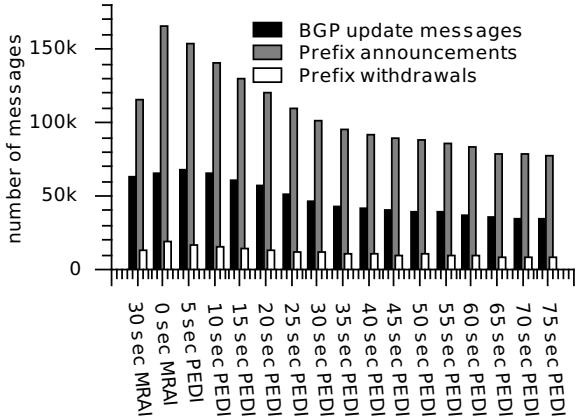


Fig. 16. Routeviews scenario: Number of BGP update messages, prefix announcements and prefix withdrawals

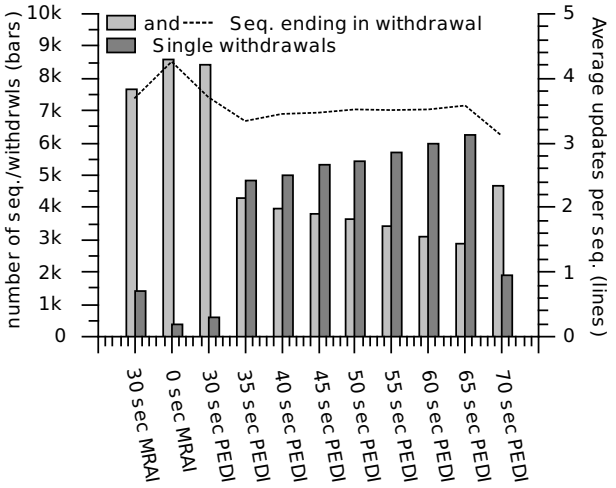


Fig. 17. APNIC scenario: Path Exploration events (update sequences ending in a withdrawal) with average updates per event and single withdrawals

second PEDI results in 18% (21151) and 32% (37441) fewer announcements respectively.

Relative to using MRAI of 30 seconds, the number of BGP update messages seen when using PED on the Routeviews dataset actually begins decreasing for PEDI values of 15 seconds and higher.

2) Impact on Path Exploration events: Next we evaluate the impact of PED and MRAI on Path Exploration events for the APNIC and Routeviews dataset.

Using a stability interval of 300 seconds, Figure 17 shows the number of Path Exploration events, and the number of single withdrawal messages[14], for each configuration of MRAI and PED using the APNIC dataset. Relative to MRAI, PED reduces the number of Path Exploration events by 44% when using a 35 second PEDI, while a 65 second PEDI reduces Path Exploration by 62%. At the same time PED increases single withdrawals by 70% with a 35 second PEDI, and by 77% with a 65 second PEDI.

Again using a stability interval of 300 seconds, Figure 18 shows the Path Exploration events and single withdrawals after applying MRAI and PED to the Routeviews dataset. As for the APNIC data, the number of single withdrawals exceeds

[14]Because PED converts Path Exploration events into single withdrawals.

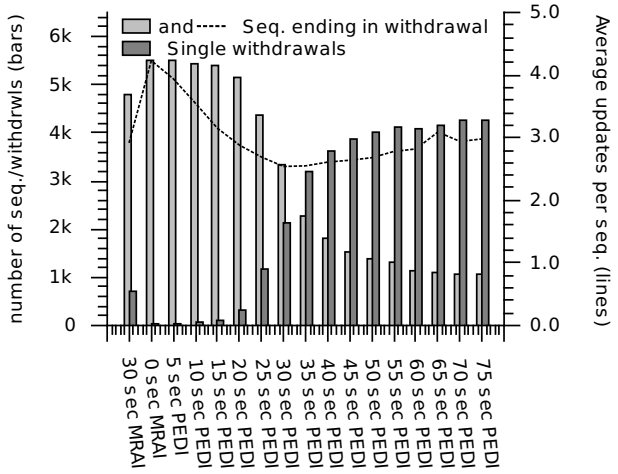


Fig. 18. Routeviews scenario: Path Exploration events (update sequences ending in a withdrawal) with average updates per event and single withdrawals

the number of Path Exploration events for PED with PEDI values higher than 35 seconds. Path exploration events begin decreasing noticeably for PEDI values of 25 seconds (by 9% (4374)) and a further 52% (2285) and 77% (1106) using 35 seconds PEDI and 65 seconds PEDI respectively, compared to using a 30 second MRAI (4789). The number of single withdrawals increases by 164%, 447% and 587% respectively from 711 to 1163, 3181 and 4172 single withdrawals. The increase of the number of single withdrawals is proportional to the decrease of the number of withdrawal sequences, showing that we are able to "clean" Path Exploration events from the BGP data, and convert them to single withdrawals as intended.

3) Impact on update sequence duration: A possible concern is the perception that PED might delay updates ending in an announcement of a longer (or equal-length) path forever[15]. We consider the possibility by analyzing the duration of update sequences ending in an announcement – either longer (or equal-length) path or shorter path.

Figure 19 shows how long the final announcement of an update sequence is delayed at Router 1. For most PEDI values this is about one PEDI interval longer than would be obtained using 30 second MRAI, giving us the impression that PED only additionally delays the last update of a sequence. The 95th percentiles strongly suggest that PED does not create update sequences which last indefinitely[16].

Figure 20 shows the durations of update sequences ending in an announcement for the Routeviews dataset. As with the APNIC data, relative to using a 30 second MRAI we see PED with PEDI of 35 seconds or higher results in a longer duration of update sequences by delaying the last update for approximately a whole PEDI period. Again the 95th percentiles strongly suggest that update sequences do not last indefinitely.

B. Potential for incremental deployment of PED

To illustrate that PED may be safely deployed in a piecewise fashion, we explore the impact on announcements and

[15]Even thought it is unlikely that such an update sequence won't be "interrupted" by a shorter path announcement at some point.

[16]In fact, the longest update sequences observed using either PED or MRAI was roughly 27K seconds, differing by only 0.2% between MRAI and PED.

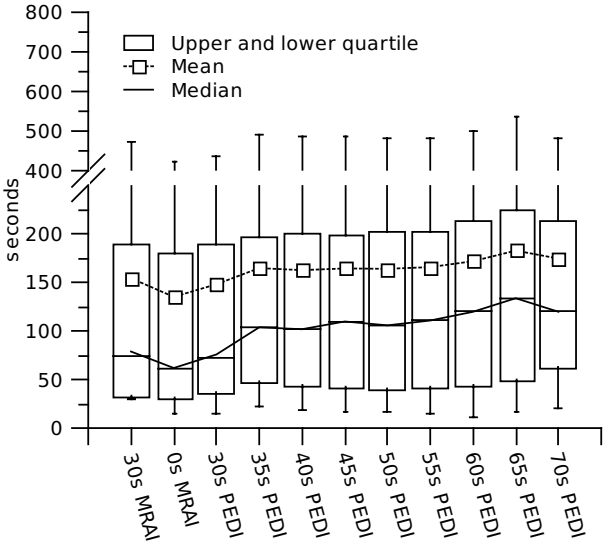


Fig. 19. APNIC scenarios: Duration of update sequences ending in announcements. Boxes bound the 25th and 75th percentiles, whiskers the 5th to 95th percentiles

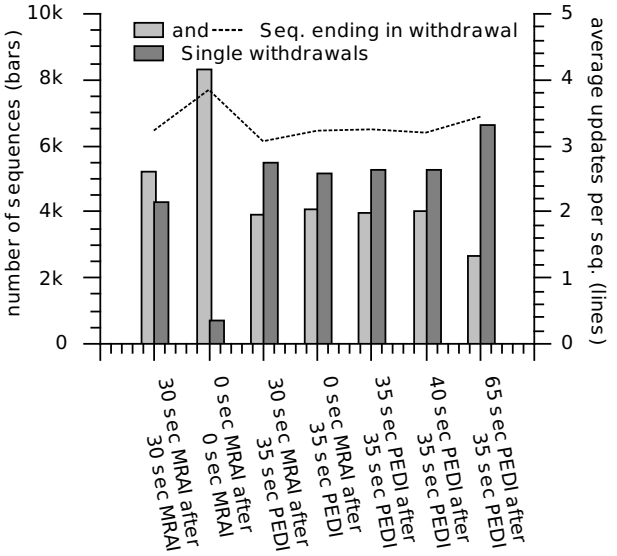


Fig. 21. APNIC: Path Exploration events (update sequences ending in a withdrawal) with average updates per event and single withdrawals downstream of Router 2 with various combinations of PED and/or MRAI at Router 1 and Router 2

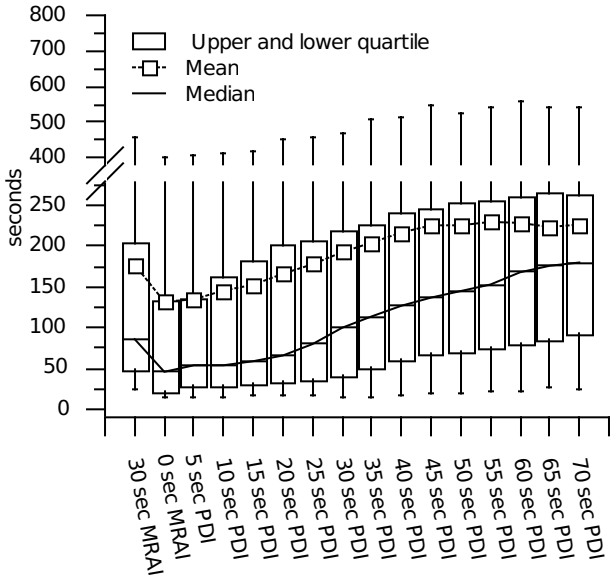


Fig. 20. Routeviews scenarios: Duration of update sequences ending in announcements. Boxes bound the 25th and 75th percentiles, whiskers the 5th to 95th percentiles

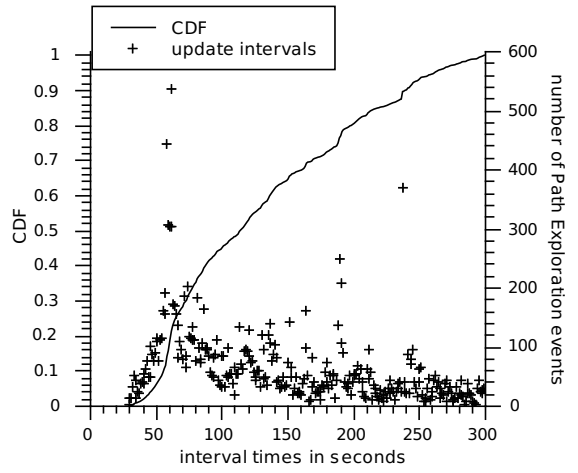


Fig. 22. APNIC: Distribution of Path Exploration intervals over 24hrs downstream of Router 1 with PED enabled. The PEDI interval at Router 1 was set to 35 seconds

updates that would, hypothetically, be emitted *further downstream* by Router 2 where Router 1 uses 0 or 30 second MRAI or 35 second PEDI, and Router 2 uses combinations of 0 or 30 second MRAI or PED with 35, 40 and 65 second PEDI.

Figure 21 shows the impact on Path Exploration events and single withdrawals observed downstream of Router 2. Application of MRAI at both routers (0 seconds or 30 seconds) results in a high number of Path Exploration events. In comparison, using 35 second PEDI first, followed by 0 seconds MRAI or 35 or 40 seconds PEDI, reduces Path Exploration events. Using 0 second MRAI, 35 second PEDI or 40 second PEDI at Router 2 all create essentially the same level of Path Exploration events. This can be explained by Figure 22, which shows there are no Path exploration events left with update intervals under 35 seconds when a PEDI of 35 seconds is employed at Router 1. (And as the next interval peak in Figure 22 is at 60 seconds, a 40 second PEDI behaves similar to a 35 second PEDI.) Compared to 0 second MRAI, 35 second PEDI or 40 second PEDI the use of 30 second MRAI at Router 2 reduces Path Exploration slightly more, as it delays random updates and interrupts some sequences. As expected, 65 second PEDI reduces Path Exploration even further.

Figure 23 shows that a router with subsequent application of PEDI creates update sequences which last slightly less than with application of a 30 second MRAI.

Consequently, it seems reasonably safe to deploy PED downstream of routers which operate a 0 or 30 second MRAI timer. This suggests incremental deployment of PED across the Internet is possible.

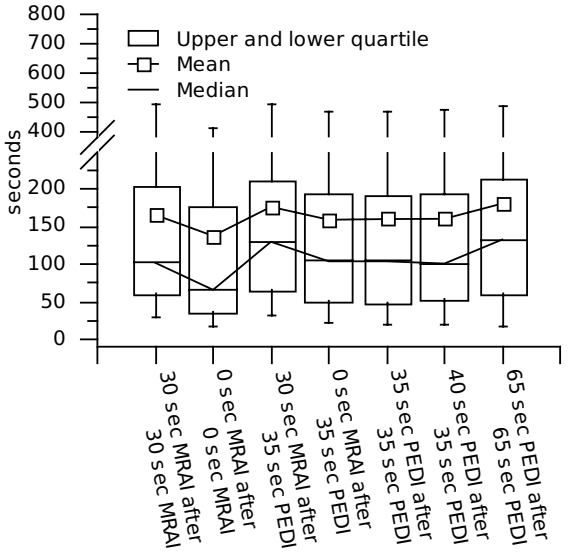


Fig. 23. APNIC: Duration of update sequences ending in announcements downstream of Router 2 with various combinations of PED and/or MRAI at Router 1 and Router 2. Boxes bound the 25th and 75th percentiles, whiskers the 5th to 95th percentiles

VIII. The impact of PED on convergence time

We now explore the relative impact of PED and MRAI on convergence time after T_{long}, T_{short}, T_{down} and T_{up} events.

A. Methodology

Using multiple instances of Quagga we constructed a routing system with the topology of Section II-D's Figure 2. For each experiment AS_1 originates the prefix 1.0.0.0/8 at $t = 0$, and we have analysed the time it takes for the initial announcement to reach the most distant AS_6, and the time it then takes for AS_6 to be notified of artificially introduced failures. We also show the various times at which reachability and optimality are achieved.

All our synthetic ASes consist of a single BGP speaker with a single configured interface, over which a BGP peering session to multiple BGP speakers is established as needed. We simulate a T_{long} and T_{short} event (as depicted in Section II-D's Figure 3) by dropping all traffic on the link between AS_{10} and AS_{11} after 300 seconds, and then allowing all traffic again 600 seconds after the simulation started.

Figure 24 shows how we simulate the T_{down} and T_{up} events. AS_1 is made temporarily unavailable for 300 seconds by disabling the IPv4 address of AS_1's BGP speaker at 300 seconds into the simulation, then re-enabling it at 600 seconds. As we are creating lossy links, rather than directly manipulating the BGP speaker, the time to register failure depends on the hold timer expiry for the directly affected BGP speakers. We use Quagga's default BGP hold timer of 180 seconds. This makes it difficult to exactly predict the failure, but allows more realistic behavior.

Reachability is considered restored after a T_{long} event when AS_{11} receives the announcement for a new route. For all other events (T_{short}, T_{down} and T_{up}), we only consider optimality, which is achieved when AS_6 receives the withdrawal in case of T_{down}, or the announcement of the route via AS_{13} in the other cases. (Optimality is also achievable before the system finally stops sending messages, such as residual announcements from AS_6 to AS_5 of an alternate path via AS_{13}.)

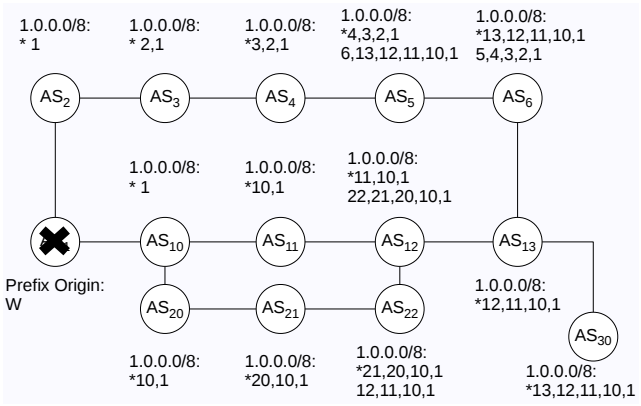


Fig. 24. T_{down} event simulated by isolating AS_1

We use a 35 second PEDI value and the common 30 second MRAI timer for each comparison. We analyze the convergence time for PED deployed on every AS, MRAI deployed on every AS, PED deployed only on AS_{12} and MRAI otherwise[17], PED deployed only on AS_{13} and MRAI otherwise and PED deployed on AS_{12} and AS_{13} with MRAI deployed on all other ASes. We have also simulated topologies with a random mixture of PED and MRAI, to verify the interaction.

B. T_{long} and T_{short} events using all PED or all MRAI

When faced with T_{long} and T_{short} events, Figures 25 and 26 illustrate a representative sequence of message arrivals at Figure 3's most relevant ASes, and changes in overall system state, when using all PED or all MRAI respectively.

With PED the initial prefix announcement took five seconds to arrive at AS_6 (as our PED implementation adds an intrinsic delay of one second per hop when propagating shorter path announcements). In comparison, the MRAI delay depends on the MRAI timer expiration, which in turn depends on the start time of both the BGP speaker and the peering sessions with its peers. In this example, MRAI took 60 seconds. However, over 20 different runs using MRAI we saw the initial announcement delayed from between 60 to 120 seconds.

On a T_{long} event we encounter a trade-off of PED: while reachability is restored almost immediately, reaching optimality is noticeably delayed. Using MRAI in the example, reachability and optimality are achieved in quick succession after the failure (with the observed delays caused by internal Quagga processing time, not by the MRAI design). Such delay also impacts on PED (for example, the initial prefix announcement arrives at AS_{12} after three or four seconds, rather than the theoretical minimum of two seconds.)

Over 20 runs using MRAI, reachability is restored between 0 and 4 seconds for 17 times, and between 29 and 30 seconds the remaining 3 times, while optimality is achieved within 1 or 2 seconds for 11 times, and between 26 and 31 seconds for 8 times. In one run MRAI delayed optimality by 58 seconds, close to the 66 seconds of PED.

[17] AS_{12} is this topology's most important *altBGP* speaker for T_{long} events

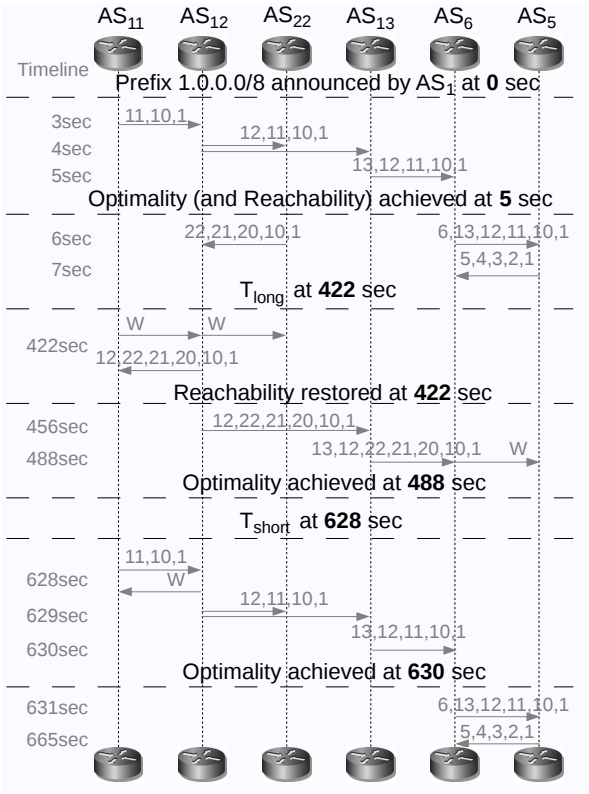


Fig. 25. Simulation of 35 seconds PEDI on all ASes: Arrival times of updates regarding prefix 1.0.0.0/8 at AS_{11}, AS_{12}, AS_{13}, AS_6, and AS_5. The time is relative to the first announcement of the prefix at AS_1. The T_{long} and T_{short} events happen on the link between AS_{10} and AS_{11}.

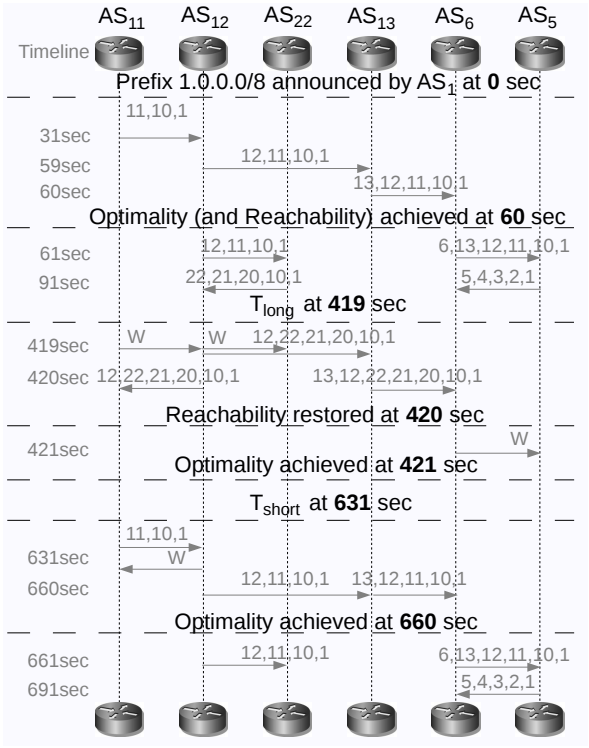


Fig. 26. Simulation of 30 seconds MRAI on all ASes: Arrival times of updates regarding prefix 1.0.0.0/8 at AS_{11}, AS_{12}, AS_{13}, AS_6, and AS_5. The time is relative to the first announcement of the prefix at AS_1. The T_{long} and T_{short} events happen on the link between AS_{10} and AS_{11}.

After a T_{short} event PED allows the new announcement to propagate without delay, and optimality is restored within two seconds, while MRAI typically delays optimality for many more seconds[18]. Over 20 runs, MRAI is observed to restore optimality mostly between 55 and 60 seconds (15 times of 20) and sometimes between 31 and 33 seconds (4 of 20). On one run MRAI even achieved optimality as quick as PED within 2 seconds. PED is more consistent than MRAI, the delay changes only by the jitter applied to longer path announcements.

C. T_{down} and T_{up} events using all PED or all MRAI

When faced with T_{down} and T_{up} events, Figures 27 and 28 illustrate a representative sequence of message arrivals at Figure 24's most relevant ASes, and changes in overall system state, when using all PED or all MRAI respectively.

The link failure is detected at different times for each peering session of AS_1, with interesting results. In the PED scenario the session between AS_1 and AS_2 is reset 45 seconds earlier than the peering session between AS_1 and AS_{10}, the AS_1 and AS_2 link failure is perceived like a T_{short} event. This causes AS_6 to send the path via AS_{13} to AS_2, which receives it three seconds after the failure.

Such a situation is suboptimal, as all the traffic from the ASes upstream of AS_6 gets redirected to AS_{13} and downstream to AS_{10}. The behavior is triggered by slow reactivity of BGP to such failures, and PED can't do anything to avoid such a situation. As soon as the link between AS_1 and AS_{10} is detected as unavailable, all the routes are withdrawn from the whole routing system immediately. After a T_{up} event, PED behaves exactly as for the initial announcement. The different times the peering sessions are restored, let PED converge twice: the first time just eight seconds after the peering session between AS_1 and AS_{10} is restored, achieving reachability, and the second time four seconds after the peering session between AS_1 and AS_2 is restored, achieving optimality.

In the MRAI scenario, the link failure is detected at different times as well, but the AS_1 and AS_{10} peering session is reset 28 seconds earlier than the AS_1 and AS_2 peering session. This is no different from the PED scenario, as the updates and withdrawals just propagate around the topology in the opposite direction. In any case, the additional updates generated by the asynchronous T_{down} event are limited to a single announcement from AS_6 to AS_{13}, announcing a path via AS_5. The MRAI timer delays that update at AS_{13} long enough for the withdrawal to arrive from AS_6, 25 seconds later and let the system converge[19]. As MRAI does not delay withdrawals, over 20 runs, every time the second link failure is detected, the withdrawals arrive at AS_6 immediately.

As for the PED scenario, the T_{up} event causes the peering sessions of AS_1 to be restored at different times. But other than for PED, even if the first peering session between AS_1 and AS_{10} is restored 23 seconds earlier than the second peering session between AS_1 and AS_2, the MRAI scenario

[18]In both cases reachability exists throughout this period – the longer path continues to carry traffic until we've optimally converged on the shorter path.

[19]It might seem, that the MRAI behavior in such a situation is better than PED, but we do not think that offloading the BGP speaker's slow reaction about a link failure to the announcement process is the correct approach.

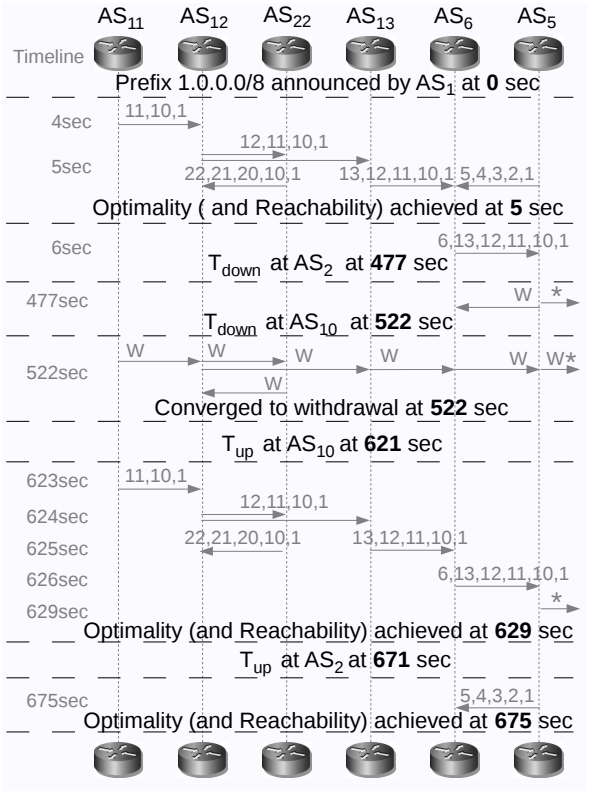


Fig. 27. Simulation of 35 seconds PEDI on all ASes: Arrival times of updates regarding prefix 1.0.0.0/8 at AS_{11}, AS_{12}, AS_{13}, AS_6, and AS_5. The time is relative to the first announcement of the prefix at AS_1. The T_{down} and T_{up} events affect the two peering sessions between AS_1 and AS_2 as well as AS_1 and AS_{10}. The * denotes updates from AS_5 to AS_4, from there to AS_3 and from there to AS_2

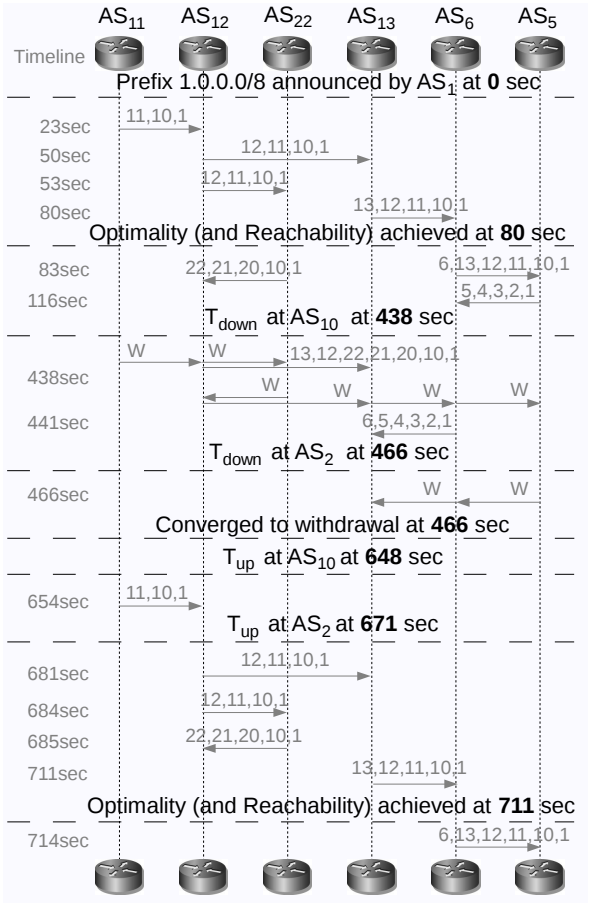


Fig. 28. Simulation of 30 seconds MRAI on all ASes: Arrival times of updates regarding prefix 1.0.0.0/8 at AS_{11}, AS_{12}, AS_{13}, AS_6, and AS_5. The time is relative to the first announcement of the prefix at AS_1. The T_{down} and T_{up} events affect the two peering sessions between AS_1 and AS_2 as well as AS_1 and AS_{10}

does not converge until the second peering session is established, and even then it takes a further 40 seconds, until AS_6 knows about prefix 1.0.0.0/8. Over 20 runs, the peering session between AS_1 and AS_{10} was randomly restored earlier or later than the session between AS_1 and AS_2. In every run updates generated by the BGP session recovering quicker, were delayed long enough to allow the updates generated by the BGP session recovering slower, to reach AS_6 almost at the same time. While this might look like a positive in terms of number of updates, it is actually a problem: Every BGP speaker upstream of AS_6 was not reachable, until the second peering session recovered. As updates to AS_6 were delayed between 76 and 90 seconds 9 times of 20, between 58 and 60 seconds 6 out of 20 and between 32 and 34 seconds 5 times of 20, PED shows a great advantage over MRAI.

D. T_{long} and T_{short} events using a mixed environment of PED and MRAI

The convergence time in a mixed environment depends on the number of PED enabled BGP speakers, and whether PED is deployed on AS_{12}. With PED deployed only at AS_{12} and MRAI on the other ASes, the initial announcement arrives at AS_6 within 46 and 90 seconds as measured over 20 test runs (The median delay is 60 seconds). After a T_{long} event, reachability is restored immediately (0 or 1 seconds over 20 runs), while optimality experiences a slightly shorter delay (between 57 and 63 seconds and 61 seconds median over 20 runs), as it would in a completely PED dominated system. After a T_{short} event, optimality can be delayed between 24 and 36 seconds and 29 seconds median over 20 runs.

Deploying PED only on AS_{13} and MRAI otherwise, the initial announcement is delayed similar to the previous scenario, between 31 and 92 seconds over 20 runs, with a median of 62 seconds. After the T_{long} event, reachability is restored with a small delay as in the all MRAI scenario (between 0 and 2 seconds) most of the time, but with the update to AS_{11} being delayed between 27 and 30 seconds in 7 out of 20 cases. while optimality is delayed by PED between 32 and 65 seconds (median of 34 seconds). Optimality after the T_{short} event is reached within 2 and 6 seconds in 4 out of 20 runs, and within 30 and 31 seconds in 16 out of 20 runs.

PED on AS_{12} and AS_{13}[20] reduces the time for the initial announcement to arrive at AS_6 6 seconds at least and 62 seconds at most (34 median over 20 runs). The T_{long} event causes AS_{12} to immediately announce the alternative route to AS_{11} restoring reachability the same moment or by a maximum delay of 2 seconds. Optimality is delayed by a PEDI period twice, once at AS_{12} and once at AS_{13} for a total delay in the range of 64 to 70 seconds (67 median). Upon the T_{short} event, optimality is restored within 2 seconds every time.

[20]This scenario is the same as for PED deployed on all routers, regarding T_{long} and T_{short} events

In any mixed PED and MRAI scenario, reachability is restored faster or equal than for the all MRAI scenario, while optimality is delayed.

E. T_{down} and T_{up} events using a mixed environment of PED and MRAI

The impact of a T_{down} and T_{up} event can be summarized for all three scenarios. After a T_{down} event, given a prevalence of ASes using MRAI, the result is similar to the all MRAI scenario – announcements generated by the first link failure are delayed at the MRAI ASes upstream of the PED ASes, and the withdrawal generated by the second link failure is propagated immediately to all ASes.

After a T_{up} event, the scenarios converge with similar times as for the initial prefix announcement described in Section VIII-D.

Using PED on AS_{12} only and on AS_{12} and AS_{13}, allowed to restore reachability while only one peering session determining the T_{up} event (either AS_1 and AS_{10} or AS_1 and AS_2) has been repaired. In the case of PED on AS_{12} only, this situation was reached 4 times out of 20, with the peering sessions being restored with only 28 seconds difference (30,59 and 64 seconds difference in the other cases). PED deployed on AS_{12} and AS_{13} showed similar results: reachability was restored 3 times out of 20, with the sessions between AS_1 and AS_{10} or AS_1 and AS_2 being restored 29, 55 and 57 seconds apart. Using PED on AS_{13} only, it was not possible to observe this behavior, even if the peering sessions were restored with 58 to 60 seconds difference 5 times out of 20. Reachability was missed by 1 hop in this case (propagation stopped either at AS_3 or AS_{11} and AS_{20} not reaching AS_2 or AS_{10} before the second peering session was restored). The behavior is completely attributable to MRAI timer synchronization.

F. Discussion

Deploying PED and MRAI randomly throughout the topology yields the following picture:

- The more PED enabled BGP speakers we have, the faster the propagation of initial announcements and achievement of optimality after a T_{short} or T_{up} event.
- If PED is deployed at AS_{12}, the *altBGP* speaker in this topology, reachability is restored immediately after T_{long}.
- The more PED is deployed, the more we experience delay in achieving optimality after T_{long}
- On a T_{down} event, in any scenario all routes are withdrawn immediately upon failure of both peering sessions.

This analysis shows that PED generally achieves reachability faster than MRAI, and that it only delays the achievement of optimality after a T_{long} event. The actual delay caused by PED depends on the PEDI timer chosen. Some have suggested lowering the standard MRAI value to 15 seconds for eBGP peers and 5 seconds for iBGP peers [25]. If such settings prevailed in the Internet PEDI could then be lowered to 20 seconds. In an Internet consisting of only PED speakers, the PEDI value could eventually be lowered further, in order to be just slightly longer than the delay experienced on withdrawals and shorter path announcements caused by BGP processing and intra AS transversal.

IX. Conclusions

Research in recent years has revealed that normal operation of BGP can amplify simple events into extended event sequences across the entire network. We consider a specific example known as Path Exploration, where a single withdrawal event can result in a superfluous series of path-lengthening announcements at intermediate routers before the withdrawal event is finally distributed across the entire network, having a negative impact on BGP convergence time.

We have implemented and demonstrated Path Exploration Damping (PED), an augmentation to BGP for selectively damping the propagation of Path Exploration updates. PED can be an alternative to the MRAI Timer.

Experimental analysis of actual BGP announcements, updates and withdrawals captured from the Internet quantify this approach and measure a reduction in the update load of up to 32% while Path Exploration events are reduced by 77%.

In terms of convergence after events, PED generally achieves reachability (a functioning forwarding path) as fast as (or faster than) MRAI regardless of the event. For path lengthening (T_{long}) events, PED does take longer than MRAI to achieve convergence to the optimal (correct) path. However, we consider the additional delay to be acceptable because data still flows (the affected prefix is reachable) while the BGP routing system attains optimal convergence.

Deployment of PED does not require a flag-day. PED provides benefits even when deployed in an incremental manner as an alternative to the MRAI Timer. PED at any one system does not require collaboration with upstream peers, and provides benefits to downstream peers who may or may not themselves implement PED.

PED is intended as a small modification to BGP but can be an alternative to methods like SSLD, Consistency Assertion, Ghost Flushing, RCN or DUP. It currently has some advantages relative each of them. Unlike SSLD, PED is compatible with the current transition to 4 byte AS numbers, compared to Consistency Assertions PED puts little overhead on the router, compared to Ghost Flushing it does not interfere with routers that still deploy RFD, compared to RCN it does not modify the protocol over-the-wire, and PED reduces the amount of updates slightly more than DUP.

Optimal values for Path Exploration Damping Interval (PEDI) depend on the mix of MRAI Timer settings used upstream of any system that deploys PED. Based on current Path Exploration statistics we propose a default PEDI of 35 seconds. A potential area for future work is to develop techniques for each BGP speaker to dynamically adjust (up or down) their local PEDI value based on observed patterns of Path Exploration events arriving from upstream peers.

X. Acknowledgments

This work has been made possible in part by a grant from the Cisco University Research Program Fund at Community Foundation Silicon Valley.

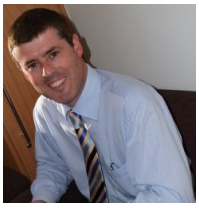
References

[1] K. Lougheed and Y. Rekhter, "Border Gateway Protocol (BGP)," RFC 1105 (Experimental), Internet Engineering Task Force, Jun. 1989, obsoleted by RFC 1163. [Online]. Available: http://www.ietf.org/rfc/rfc1105.txt

[2] Y. Rekhter, T. Li, and S. Hares, "A Border Gateway Protocol 4 (BGP-4)," RFC 4271 (Draft Standard), Internet Engineering Task Force, Jan. 2006. [Online]. Available: http://www.ietf.org/rfc/rfc4271.txt
[3] G. Huston, "BGP Routing Table Resource Pages," June 2009. [Online]. Available: http://bgp.potaroo.net
[4] D. Meyer, L. Zhang, and K. Fall, "Report from the IAB Workshop on Routing and Addressing," RFC 4984 (Informational), Internet Engineering Task Force, Sep. 2007. [Online]. Available: http://www.ietf.org/rfc/rfc4984.txt
[5] Y. Rekhter and T. Li, "A Border Gateway Protocol 4 (BGP-4)," RFC 1771 (Draft Standard), Internet Engineering Task Force, Mar. 1995, obsoleted by RFC 4271. [Online]. Available: http://www.ietf.org/rfc/rfc1771.txt
[6] C. Labovitz, A. Ahuja, A. Bose, and F. Jahanian, "Delayed internet routing convergence," *IEEE/ACM Trans. Netw.*, vol. 9, no. 3, pp. 293–306, 2001.
[7] C. Villamizar, R. Chandra, and R. Govindan, "BGP Route Flap Damping," RFC 2439 (Proposed Standard), Internet Engineering Task Force, Nov. 1998. [Online]. Available: http://www.ietf.org/rfc/rfc2439.txt
[8] Z. M. Mao, R. Govindan, G. Varghese, and R. H. Katz, "Route flap damping exacerbates internet routing convergence," in *SIGCOMM '02: Proceedings of the 2002 conference on Applications, technologies, architectures, and protocols for computer communications*. New York, NY, USA: ACM, 2002, pp. 221–233.
[9] T. Li and G.Huston, "BGP Stability Improvements," Internet-Draft (Informational), Jun. 2007. [Online]. Available: http://tools.ietf.org/html/draft-li-bgp-stability-01
[10] P. Smith and C. Panigl, "RIPE Routing Working Group Recommendations on Route-flap Damping," ripe-378, May 2006, obsoletes: ripe-229, ripe-210, ripe-178. [Online]. Available: http://www.ripe.net/docs/ripe-378.html
[11] T. Griffin and B. Premore, "An experimental analysis of BGP convergence time," in *Network Protocols, 2001. Ninth International Conference on*, Nov. 2001, pp. 53–61.
[12] J. Chandrashekar, Z. Duan, Z.-L. Zhang, and J. Krasky, "Limiting path exploration in BGP," in *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, vol. 4, March 2005, pp. 2337–2348 vol. 4.
[13] R. Oliveira, B. Zhang, D. Pei, R. Izhak-Ratzin, and L. Zhang, "Quantifying path exploration in the internet," in *IMC '06: Proceedings of the 6th ACM SIGCOMM conference on Internet measurement*. New York, NY, USA: ACM, 2006, pp. 269–282.
[14] G. Huston, "ISP column: Path Damping," June 2007. [Online]. Available: http://www.potaroo.net/ispcol/2007-06/dampbgp.html
[15] D. Pei, X. Zhao, D. Massey, and L. Zhang, "A Study of BGP Path Vector Route Looping Behavior," in *ICDCS '04: Proc. 24th International Conference on Distributed Computing Systems (ICDCS'04)*. Washington, DC, USA: IEEE Computer Society, 2004, pp. 720–729.
[16] C. Labovitz, A. Ahuja, and F. Jahanian, "Experimental study of internet stability and backbone failures," *Fault-Tolerant Computing, 1999. Digest of Papers. Twenty-Ninth Annual International Symposium on*, pp. 278–285, 1999. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=781062
[17] C. Labovitz, A. Ahuja, R. Wattenhofer, and S. Venkatachary, "The impact of internet policy and topology on delayed routing convergence," in *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 1, 2001, pp. 537–546 vol.1.
[18] D. Pei, X. Zhao, L. Wang, D. Massey, A. Mankin, S. Su, and L. Zhang, "Improving BGP convergence through consistency assertions," in *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 2, 2002, pp. 902–911 vol.2.
[19] A. Bremler-Barr, Y. Afek, and S. Schwarz, "Improved BGP convergence via ghost flushing," in *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, vol. 2, March-3 April 2003, pp. 927–937 vol.2.
[20] D. Pei, M. Azuma, D. Massey, and L. Zhang, "BGP-RCN: improving BGP convergence through root cause notification," *Comput. Netw. ISDN Syst.*, vol. 48, no. 2, pp. 175–194, 2005.
[21] W. Sun, Z. Mao, and K. Shin, "Differentiated BGP Update Processing for Improved Routing Convergence," in *Network Protocols, 2006. ICNP '06. Proceedings of the 2006 14th IEEE International Conference on*, Nov. 2006, pp. 280–289.
[22] M. Rossi, "Implementing path-exploration damping in the Quagga Software routing Suite Version 0.99.13 - patch set version 0.3," Centre for Advanced Internet Architectures, Swinburne University of Technology, Melbourne, Australia, Tech. Rep. 090730A, 30 July 2009. [Online]. Available: http://caia.swin.edu.au/reports/090730A/CAIA-TR-090730A.pdf
[23] K. Ishiguro, "Quagga Software Routing Suite." [Online]. Available: http://www.quagga.net
[24] M. Rossi, "Quagga-Accelerator: An Implementation for Accelerated Processing of Historical BGP Events using Quagga 0.99.13 - version 0.1," Centre for Advanced Internet Architectures, Swinburne University of Technology, Melbourne, Australia, Tech. Rep. 090730C, 30 July 2009. [Online]. Available: http://caia.swin.edu.au/reports/090730C/CAIA-TR-090730C.pdf
[25] P. Jakma, "Revised Default Values for the BGP 'Minimum Route Advertisement Interval'," draft-jakma-mrai-02.txt (Internet Draft), November 2008. [Online]. Available: http://tools.ietf.org/html/draft-jakma-mrai-02

Geoff Huston holds a B.Sc. and a M.Sc. from the Australian National University. He has been closely involved with the development of the Internet for many years, particularly within Australia, where he was responsible for the initial build of the Internet within the Australian academic and research sector. He is currently the Chief Scientist at APNIC, the Regional Internet Registry serving the Asia Pacific region. He is a longstanding member of the Internet Engineering Task Force, a member of the Internet Architecture Board from 1999 until 2005, and served on the Board of the Internet Society from 1992 until 2001.

Mattia Rossi holds a B.Eng. and a M.Sc. (Dipl.Ing.) from the Leopold-Franzens-Universitaet Innsbruck, Austria. He is currently working as Research and Development Engineer at the Centre for Advanced Internet Architectures at Swinburne University of Technology, Melbourne, Australia. He has been involved in transport layer and network security research, and is performing BGP, routing and network layer related research since 2008.



Grenville Armitage earned a B.Eng (Elec)(Hons) in 1988 and a PhD in electronic engineering in 1994, both from the University of Melbourne, Australia. He is currently Professor of Telecommunications Engineering and Director of the Centre for Advanced Internet Architectures at Swinburne University of Technology, Melbourne, Australia. He authored Quality of Service In IP Networks: Foundations for a Multi-Service Internet (Macmillan Technical Publishing, April 2000) and co-authored Networking and Online Games - Understanding and Engineering Multiplayer Internet Games (John Wiley Sons, UK, April 2006). Professor Armitage is also a member of ACM and ACM SIGCOMM.