Route Flap Damping Made Usable

Cristel Pelsser[1], Olaf Maennel[2], Pradosh Mohapatra[3], Randy Bush[1], and Keyur Patel[3]

[1] Internet Initiative Japan
Tokyo, Japan
{cristel,randy}@iij.ad.jp

[2] Loughborough University
United Kingdom
O.M.Maennel@lboro.ac.uk

[3] Cisco Systems
San Jose, CA, USA
{pmohapat,keyupate}@cisco.com

Abstract. The Border Gateway Protocol (BGP), the de facto inter-domain routing protocol of the Internet, is known to be noisy. The protocol has two main mechanisms to ameliorate this, MinRouteAdvertisementInterval (MRAI), and Route Flap Damping (RFD). MRAI deals with very short bursts on the order of a few to 30 seconds. RFD deals with longer bursts, minutes to hours. Unfortunately, RFD was found to severely penalize sites for being well-connected because topological richness amplifies the number of update messages exchanged. So most operators have disabled it. Through measurement, this paper explores the avenue of absolutely minimal change to code, and shows that a few RFD algorithmic constants and limits can be trivially modified, with the result being damping a non-trivial amount of long term churn without penalizing well-behaved prefixes' normal convergence process.

1 Introduction

Despite the huge success of the Internet, the dynamics of the critically important inter-domain routing protocol, the Border Gateway Protocol (BGP), remain a subject of research. In particular, despite a large number of research efforts, the convergence of BGP [6, 11], and lately, the chattiness of BGP, also called BGP churn [3], are still not well understood. Further observations have been made of duplicated and/or 'unnecessary' updates [16]. These all ultimately lead to slow protocol convergence.

Understanding the BGP mystery is critical. In the case of convergence, vendors may improve code based on insights into propagation patterns, which in turn could lead to less churn, and thus lower load, a more robust network, and faster response to failure events. Researchers suggesting replacement protocols could design them with an in-depth understanding of what works today, what does not work well, and why.

This paper aims at one facet in this spectrum: how, with absolutely minimal code change, to better differentiate the *normal* path-vector protocol convergence process from *abnormal* activity, such as heavily flapping prefixes. It has been shown that a

single triggering event can cause multiple BGP updates elsewhere in the Internet [5,6]. We say a BGP route is flapping or unstable if a router *originates* multiple BGP update messages (reachable or unreachable) for the prefix in a 'short' time interval and propagates those changes to its neighbors. However, BGP, being a path-vector protocol is also subject to *topological amplification*, sometimes called *path exploration*. One triggering event can cause multiple BGP updates at a topologically distant router. Studies using BGP beacons [13] have illustrated this effect. It is important to understand that this is a property (or artifact) of the BGP protocol itself and does not correspond to constantly changing topology. In fact, studies of BGP update behavior and traffic flow have found little correlation [21]. The traffic may continue to reach its destination despite the constant noise of BGP update messages.

While this is conceptuahly very simple, it is not easy to distinguish real topological changes from path exploration in the BGP signal. Ideally, we would like to maximize the speed topological information is propagated, while minimizing exchanged messages required to *converge* to a stable path. However, the root cause of a BGP update typically cannot be known. Therefore mechanisms to reduce BGP's chattiness face the dilemma of finding appropriate algorithms and parameters.

Huston [8] has observed that a small portion of the prefixes generate a high number of BGP update messages. In Figure 1 we show a similar observation. Most prefixes receive very few updates. Only 3% of the prefixes are responsible for 36% percent of the BGP messages. The plot shows the number of update messages that are received at a router in our measurement setup (Fig. 3) for each prefix during the week from Sept. 29th to Oct. 6th, 2010.
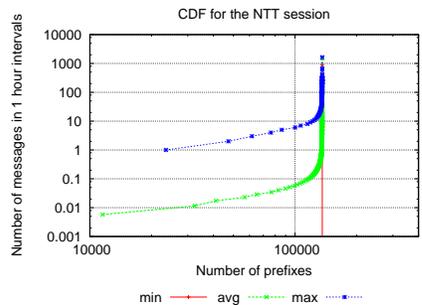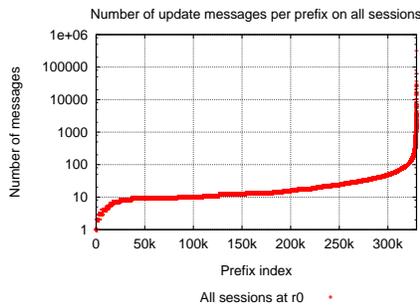

Fig. 1. Update count per prefix at *r*0 during the week of Sept. 29th to Oct. 6th 2010.

Fig. 2. Update count per prefix from a single BGP session in one hour bins.

Figure 2 illustrates the churn, i.e. update messages per hour that are received on a session with a tier-1 ISP. The y-axis depicts the number of updates received for a particular prefix per one hour bin, while the x-axis shows the prefixes sorted by the number of update messages received. The majority of prefixes account for few updates, while a small number of prefixes account for a very high number of updates within a short time period. The figure shows three curves, the minimum (vertical line), the average (lower curve) and maximum (top curve) number of updates in one hour bins.

Router $r0$ receives a full routing table, $326,575$ routes, from NTT. One might expect that most of those routes would be stable and not receive any updates at all. However, we observe updates for $153,773$ prefixes during *one week* of observation. And the router receives up to $1,647$ updates in one hour for the prefix with the highest churn (see right most point on the top curve in Figure 2), there are less than ten updates for more than $100,000$ of the prefixes for which there were any updates. Most prefixes for which we observe BGP update messages are quiet most of the time. Only 0.01% of the prefixes are always present in the trace, with one prefix having a minimum of 913 BGP updates per hour over the whole trace (which explains the vertical line in Figure 2). These observations confirm that most prefixes are very quiet, and only a very small number of the prefixes are responsible for the majority of the BGP churn.

For some prefixes the router received hundreds and thousands of update messages, over arbitrarily long time-periods. We hypothesize those updates are being caused by some periodic events and/or flapping. This cannot be 'normal' protocol convergence. This is causing an unnecessary load on the global routing system.

2 Background

There are many causes for route flapping. One common cause is a router or a link going up and down due to a faulty circuit or hardware. Another cause is a BGP session being reset. BGP policy changes can also lead to the readvertisement of routes and can thus be interpreted as a route flap, this also includes policy changes for traffic engineering. Furthermore, IGP cost changes may cause BGP updates which then propagate across the Internet [18]. Duplicate advertisements [16] are probably the best example of 'unnecessary' updates that do not contain any new topological information. Lastly, the BGP protocol is known to be inherently unstable [1,4,7].

Today, two approaches attempt to make the trade-off between convergence time and message count [6]. First, the MinRouteAdvertisementInterval timer (MRAI) [17] specifies the minimum time between BGP advertisements to a peer. While it is recommended to be a per prefix timer, existing implementations typically use a per-peer timer for all prefixes sent via that peering. By default, it is 30 seconds (jittered) for an eBGP peer, and five seconds for iBGP. The idea is that the router waits for the 'path exploration' downstream to finish, before sending any updates. However, as mentioned earlier, no technique can reliably discriminate between flapping routes and routes that are 'converging'.

The second technique is Route Flap Damping (RFD) [20]. It is more complex and fine-grained, as routers maintain a penalty value per prefix and per session. Routes with a penalty above a given threshold are damped, e.g., newly received announcements are suppressed and not considered as suitable alternatives to reach a destination. The idea is that heavily flapping paths are putting a large burden on the routing system as a whole and to protect the Internet from such routes, it is better to disregard the path and drop its traffic than to let such prefixes potentially cause cascading failures due to system overload. Of course, despite observations, stable routes are not supposed to be affected by this mechanism. Thus, there is still room for research in this area. For instance, the work of Huston [10] is promising in that it aims to categorize updates and determine

the types that are potential indicators of path hunting. However, live detection of such updates is much more CPU and memory intensive than the brutally simple approach explored in this paper.

Using RFD [20], each prefix accumulates a penalty which is incremented on receipt of an announce or withdraw message for that prefix. This penalty is a simple counter and the values added to the penalty are listed in Table 4. When the penalty reaches a given threshold, the 'suppress penalty', the route is damped, i.e. quarantined. It is not advertised by the router until the penalty gets below another threshold, the 'reuse penalty'. The penalty value of a damped route is decremented using a 'half-life', i.e. it is divided by two after 'half-life' seconds. Upon the receipt of further updates the penalty continues to grow. However, there is a 'max suppress time', which constitutes a maximum time the route can be damped. E.g., provided that the route is not receiving any further updates, a damped prefix is typically released after one hour. This translates into a 'maximum suppress penalty', which is computed using the suppress threshold, the reuse threshold and the half-life time. For example, with Cisco default parameters a penalty of 12,000 will result in a suppression of one hour if no further updates for that prefix arrive. We refer to the work of Mao et al. [14] for a detailed study of the RFD algorithm.

RFD has been reported to be harmful [2] in that, with current default settings and recommendations [15], it penalizes routes which are not flapping, but receiving multiple updates due to path exploration. This severely impacts convergence. Reachability problems for over an hour have been observed where there was no physical outage, network problem, or congestion that would justify any packet drops [12]. In fact, it has been shown that perfectly valid and fine paths can be withdrawn due to RFD [2]. As a consequence most operators have disabled RFD. On the other hand, we see serious BGP noise affecting router load and burdening the whole system [9].

Can research on BGP dynamics lead to an appropriate recommendation of RFD parameters? What would happen if we adopted a strategy to select only the 'heavy hitters', the heavily flapping routes, or 'elephants' as we call them – but leave the converging routes, or 'mice', in peace? BGP churn should decrease significantly compared to the current situation where RFD is turned off, yet the BGP convergence for prefixes with 'normal' BGP activity would not be affected. In this paper, we try to find and propose such appropriate parameters.

3 Measurement Setup

In this section, we present our experimental design. We describe a change to Cisco's IOS XR BGP implementation to enable the collection of damping statistics, the location of the router in the Internet and the BGP feeds that it receives. Then we explain how we collected and analyzed the RFD data.

Router *r0* in Figure 3 is a Cisco 12406 running a minimally modified version of Cisco's IOS XR software to enable us to perform a detailed analysis of what the router 'thinks'. The router applies the RFD algorithm using the normal penalty values. The modified code does not actually damp the routes, instead it records the calculated penalty values of each route and its supposed status, active or damped. The other modi-
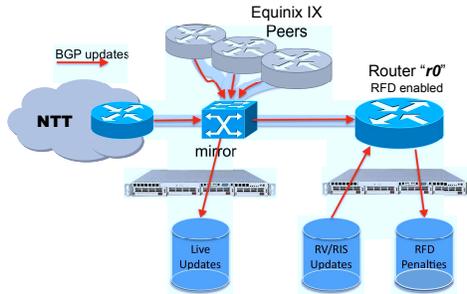
Fig. 3. Measurement Topology Setup.

	Parameter	Value
1	Half-life time	15 min
2	Max suppress penalty	12,000
3	Max suppress time	60 min
4	Suppress penalty	2,000
5	Reuse penalty	750
6	Withdrawal penalty	1,000
7	Re-advertisement penalty	0
8	Attribute change penalty	500

Fig. 4. Cisco's default RFD values

fication was that no 'Maximum Suppress Penalty' was imposed, e.g., the penalty values could increase above 12,000.

Figure 3 shows our measurement infrastructure. Router *r0* is directly connected to a large public Internet Exchange over which it receives both full and partial feeds. In addition, the router connects to a global tier-1 provider for another full BGP feed.

We pulled statistics from the router at regular intervals for one week, from September 29 through October 6, 2010, using the clogin command from the rancid tool. Data included details of all route flap damping counters, although the router code did not actually damp any route. The time to pull the data from the router depended on how quickly the router responded to our queries, but was typically in the order of 4–5 minutes. Missing counter values due to slow router response time did not significantly affect our observations in subsequent sections, as there were very few of them. The 95% quantile was under ten minutes. However, in some circumstances it was longer, up to 45 minutes in one instance! We believe this was due to CPU utilization peaks.

4 Results

We investigate the penalty values assigned to the prefixes received by our modified router, $r0$ (Figure 3). We then provide recommendations for new RFD parameter settings.

Figure 5 shows the Cumulative Distribution Function (CDF) of the penalties assigned to prefixes by the router during the one week experiment. Let us assume there are n snapshots during the week's experiment. We define an 'instance' $i_{p,t}$ as the RFD penalty of prefix p in snapshot t. Figure 5 shows the proportion of instances with penalties smaller than or equal to x over the whole set of instances. Intuitively, this is the proportion of prefixes which would have been damped in the time-prefix-space.

We observe that 14% percent of the instances reached a penalty greater or equal to 2,000 in the measurement period. 2,000 is a critical threshold as this is the default value for RFD suppression on Cisco routers. This gives a feeling for how 'bad' it is, if one turns on default RFD those instances would have been damped. Further, we observe in
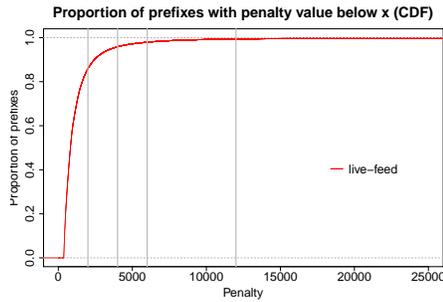
Fig. 5. Distribution of penalty values. Vertical lines are 2,000, 4,000, 6,000, and 12,000.
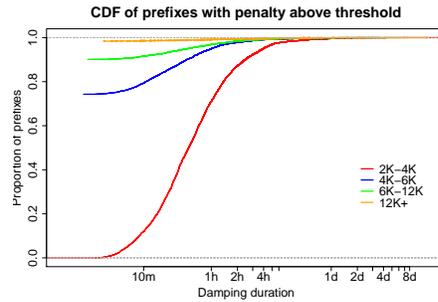

Fig. 6. CDF of the proportion of prefixes with penalty values above a threshold.

Figure 5 that a suppress threshold of 4,000, 5,000 and 6,000 leads to the damping of 4.2%, 2.8% and 2.1% of the instances respectively. The number of damped instances decreases very quickly. Finally, we note that very few of the prefixes are assigned a very high penalty. Only 0.63%, 0.44% and 0.32% have a penalty value above 12,000, 15,000 and 18,000, respectively. Thus, very few prefixes flap heavily for long in the time-prefix-space. However, we observed earlier that those few prefixes are responsible for a disproportionate part of the BGP churn. The maximum penalty value assigned to a route during the experiment was 48,000. This value is huge compared to the median penalty of 818 (Fn(818)=0.5).

We recommend conservative operators set the 'suppress threshold' to 12,000, 15,000 or 18,000, as these values likely penalize only the very heavy hitters. We show later that, while values in the range [12,000 − 18,000] enable a non negligible BGP update rate reduction, a suppress threshold in the range [4,000 − 6,000] damps far fewer prefixes compared to current defaults and the BGP update rate is significantly reduced.

How long do prefixes typically stay at high penalty values? Figure 6 shows the CDF of the durations a prefix is above a certain penalty value, and thus would be damped if this was the threshold. The red solid curve shows the damping duration for the current threshold of 2,000. Many prefixes have a penalty above 2,000 for a very short time. For example, 68% of prefixes stay above 2,000 for up to one hour during the one week of the experiment. This means the current default suppresses a lot of prefixes that are unstable for a relatively short time. We suspect that many of those prefixes are inappropriately damped following a single event. They are given a penalty value above 2,000 during BGP convergence simply because of path exploration. We should not damp those prefixes!

The other curves show suppression times for penalty values between 2K and 4K, between 4K and 6K, 6K–12K, and above 12K relative to those prefixes in the 2K class. If a prefix is not suppressed at all, then the duration is zero and thus the curve starts at this point on the y-axis. Not surprisingly, the number of prefixes in each category varies quite a lot (721 prefixes above 12K, top most curve; 4,429 prefixes between 6K–12K, 2nd from top; and 11,546 prefixes between 4K–6K, 3rd from top; and 44,846 prefixes between 2K–4K, lowest curve). Furthermore, there are very few prefixes that have a high penalty for a long time (e.g. rightmost points). There are 57 prefixes in the 2K–4K

band that stay in this band for more than two days, but only 12 prefixes in the above 12K-band that stay for more than two days. We noticed some prefixes change bands, e.g., stay for a few hours/days in the 2K–4K band and then also stay a few hours/days in a higher band. Overall, it is possible that high churn prefixes stay for quite some time in lower bands; but we have also shown that 'normal converging' prefixes stay in those bands. Therefore, we need to find a trade-off in the parameter space, that does not penalize prefixes that only experience path exploration.

Figure 7 shows the number of prefixes which would be damped given the different candidate thresholds. Clearly, (32,089) mice would be spared using a suppress threshold of 4,000 or above. Moreover, we see that the number of prefixes damped with higher suppress thresholds does not vary much. High thresholds are much more suitable to prevent damping of prefixes affected by normal BGP path exploration than the current default threshold. Our intuition here is that a 'badly behaving prefix' will flap for a long time and thus hit high penalty values; while 'normal converging prefixes', which just receive multiple updates due to path exploration, will not be penalized.
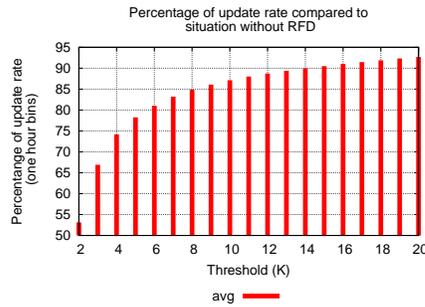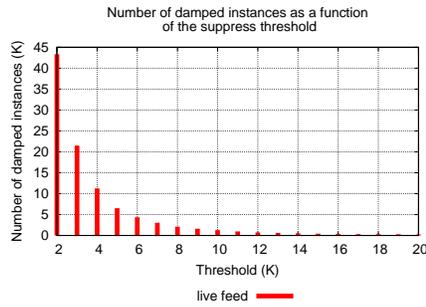

Fig. 7. A suppress threshold of 4,000 (and above) already damps many fewer prefixes.

Fig. 8. A threshold of 6,000 enables an average churn reduction of 19%.

Increasing the value of the suppress threshold above today's default will increase the BGP churn rate, but it will save many mice and still be less churn than a router with RFD turned off. Figure 8 illustrates this. The x-axis is the candidate value of the suppress threshold. On the y-axis we show the update rate on a per minute average in 60 minute bins. 100% is the churn when RFD is disabled.

Here, we try to estimate how churn could change if we activate RFD at various thresholds. Unfortunately, we face two problems: (1) the accuracy of timestamps and (2) we are looking at only one router and not studying interactions in complex topologies. With regard to (1) we record incoming updates via tcpdump with sub-second accuracy. However, the router provides us with less frequent snapshots of the penalty values. We therefore have only an estimate of the penalty. Especially, we do not know the exact time of onset, e.g., when an update would have been damped. Updates often come in bursts with short inter-arrival times but the arrival process of bursts is rather uniformly distributed in time, and thus within the snapshots. If a prefix has a penalty above the considered threshold in the current snapshot, all its updates in the coming interval are

marked as being potentially removed. This provides an estimation of the update rate. By averaging over the whole trace, the error smooths out. We tag all updates that cross our 2K, 3K,... thresholds within a certain time-window. With respect to (2) we cannot predict how MRAI and best path selection processes will or will not delay updates. While we believe that the overall properties of update behavior are comparable, we leave it for future work to study the impact in complex topologies.

We observe a 47% reduction of the average update rate with a penalty of $2,000$, compared to a situation without RFD, in Figure 8. $4,000$, $5,000$ and $6,000$ correspond to an average update rate reduction between 26% and 19%. Thus, it is worthwhile changing the default suppress threshold value. Our proposal is a very simple modification which is rather effective compared to more complex solutions such as [10].

We further note that the churn reduction is similar for all thresholds above 12K. Damping thresholds of 12K, 15K and 18K suppress an average of 11.26%, 9.51% and 8.12% of the updates, which is still non negligible for such a trivial change as we propose.

To compare the really heavy hitters in the intervals 12K-15K, 15K-18K, and above 18K, we concentrate on 64 prefixes which have a damped duration of six hours or longer. We notice that 53 of those 64 prefixes (83%) at some point pass the high point of 18K. Only nine prefixes (14%) stay in the 12K-15K range, and only two prefixes (3%) go over 15K, but not up to 18K. This strengthens our confidence that the 'evil' guys, the really heavy hitters which constantly flap, will be caught by almost any threshold setting, be it 12K, 15K, or 18K.

Thus, for more conservative operators that desire to spare most of the mice and still see around 10% churn reduction, we recommend values 12K and above. It does not matter much which of these three values are chosen. If a prefix is flapping so hard that it reaches 12K, then it is also likely to go higher at some time.

5   Other feeds

A critical question is whether the observations and recommendations in the previous section hold for other locations in the Internet topology? Can we make a generic recommendation for the 'suppress penalty' value? To understand this, we replayed additional varied BGP traces from Route Views into $r0$ (see 'RV/RIS Updates' in Figure 3) in pseudo real-time. Again, $r0$ logged the RFD penalties of the received routes.

We performed two additional experiments. Figure 10 describes the additional workload traces that were replayed to the router. These were in addition to the in vivo feeds from the tier-1 ISP and the Exchange Point.

These experiments were designed to determine if different update patterns recorded at different places in the Internet topology would affect our conclusions.

Figure 9 shows the penalty values for prefixes with the new feeds replayed from Route Views. It shows that the distributions are exceedingly similar to those from the live feeds. Similar to Figure 5, this plot shows a CDF of penalties assigned by $r0$ to the different instances in the time-prefix-space. The green curve is the workload from the live feed plus a BGP feed from an African peering point. The blue curve is the 1.5 day live workload with the 10 additional Route Views feeds. The red curve is the one week
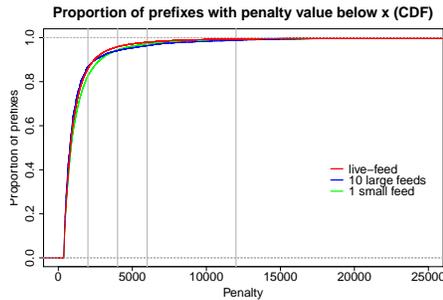
Proportion of prefixes with penalty value below x (CDF)

live−feed
10 large feeds
1 small feed

10 large feeds: Ten Route Views [19] feeds which received the maximum number of updates from August 5 to mid-day August 6, 2010 (1.5 days).

1 small feed: A small selected African Exchange, route-views.kixp. Data were collected from August 27 to September 1, 2010 (5 days).

Fig. 9. Distribution of penalty values.

Fig. 10. Additional feeds.

workload (live feed), previously shown (see Figure 5) for comparison. We observe that all three curves have a similar shape. Adding more feeds just leads to more prefixes that flap but the number of 'elephants' is very similar.

Therefore, our damping suppression threshold recommendation does not change for BGP feeds from varying points in the Internet topology.

6   Conclusion

We studied the impact of RFD on the Internet. As previously observed by many other researchers, a small fraction of prefixes is responsible for a significant portion of the update churn. RFD was developed to reduce this noise, but the current default parameters do not properly take into account properties of the BGP protocol. Any path-vector protocol is by it's very design noisy due to *path exploration.*

We therefore looked at the effect of an absolutely minimal change, only adjusting RFD parameters, to get moderate churn reduction without adversely impacting normally converging prefixes. Our recommendation derived from this study would be to damp a route when it reaches a penalty above $12,000$. The suppress threshold can be set to any value between $12,000$ and $18,000$. Such a setting will suppress the BGP churn of routes that flap heavily while keeping paths for prefixes which only slightly contribute to BGP churn. For operators extremely concerned about churn, a suppress threshold of $4,000$ to $6,000$ is a far better compromise than today's default parameters. It may still damp some normally converging prefixes, but will also significantly reduce the BGP update rate.

We do not recommend changing the maximum suppress time, but we strongly recommend the limit of the maximum suppress threshold value be raised. A maximum suppress time of one hour is very reasonable to achieve recovery once the flapping stops (and heavy hitters will anyway broadcast continuously), but the maximum suppress threshold needs to be able to allow higher values than $12,000$.

Acknowledgments

and Equinix contributed significant support. We are thankful to Matthew Roughan for many comments on earlier versions of this idea. We would also like to thank Nate Kushman for the inspiring discussions.

References

1. Basu, A., Ong, C.H.L., Rasala, A., Shepherd, F.B., Wilfong, G.: Route Oscillations in I-BGP with Route Reflection. In: Proc. ACM SIGCOMM (2002)
2. Bush, R., Griffin, T., Mao, Z.M.: Route flap damping: harmful? RIPE 43 (2002), `http://www.ripe.net/ripe/meetings/archive/ripe-43/presentations/ripe43-routing-flap.pdf`
3. Elmokashfi, A., Kvalbein, A., Dovrolis, C.: BGP churn evolution: A perspective from the core. In: In Proceedings of INFOCOM (2010)
4. Griffin, T.G., Wilfong, G.: Analysis of the MED Oscillation Problem in BGP. In Proceedings of the International Conference on Network Protocols (2002)
5. Griffin, T.G.: What is the Sound of One Route Flapping? (2002), IPAM
6. Griffin, T.G., Premore, B.J.: An Experimental Analysis of BGP Convergence Time. In: Proc. ICNP (2001)
7. Griffin, T.G., Wilfong, G.: On the correctness of iBGP configuration. SIGCOMM Comput. Commun. Rev. 32(4), 17–29 (2002)
8. Huston, G.: The BGP instability report (2006), `http://bgpupdates.potaroo.net/instability/bgpupd.html`
9. Huston, G.: BGP Extreme Routing Noise. RIPE 52 (2006), `http://www.ripe.net/ripe/meetings/ripe-52/presentations/ripe52-plenary-bgp-review.pdf`
10. Huston, G.: Update damping in BGP (2007), `www.potaroo.net/presentations/2007-10-25-dampbgp.pdf`
11. Labovitz, C., Ahuja, A., Bose, A.: Delayed Internet Routing Convergence. In Proceedings of SIGCOMM pp. 175–177 (August 2000)
12. Mao, M., Govindan, R., Varghese, G., Katz, R.: Route Flap Damping Exacerbates Internet Routing Convergence. In: Proc. ACM SIGCOMM (2002)
13. Mao, Z.M., Bush, R., Griffin, T.G., Roughan, M.: BGP Beacons. In: Proc. ACM IMC (2003)
14. Mao, Z.M., Govidan, R., Varghese, G., Katz, R.H.: Route Flap Damping Excacerbates Internet Routing Convergence. In Proceedings of SIGCOMM (August 2002)
15. Panigl, C., Schmitz, J., Smith, P., Vistoli, C.: RIPE Routing-WG Recommendation for Coordinated Route-flap Damping Parameters (2001), `http://www.ripe.net/ripe/docs/ripe-229.html`
16. Park, J.H., Jen, D., Lad, M., Amante, S., McPherson, D., Zhang, L.: Investigating occurrence of duplicate updates in BGP announcements. In: Proc. Passive and Active Measurement Workshop (PAM) (2010)
17. Rekhter, Y., Li, T.: A Border Gateway Protocol 4 (BGP-4) (2006), RFC 4271
18. Teixeira, R., Shaikh, A., Griffin, T.G., Voelker, G.M.: Network sensitivity to hot-potato disruptions. In: Proc. ACM SIGCOMM (2004)
19. University of Oregon RouteViews project: `http://www.routeviews.org/`
20. Villamiyar, C., Chandra, R., Govindan, R.: BGP route flap damping (1998), RFC 2439
21. Wang, F., Mao, Z.M., Wang, J., Gao, L., Bush, R.: A Measurement Study on the Impact of Routing Events on End-to-End Internet Path Performance. In: Proc. ACM SIGCOMM (2006)