# Investigating occurrence of duplicate updates in BGP announcements

Jong Han Park[1], Dan Jen[1], Mohit Lad[2], Shane Amante[3], Danny McPherson[4], and Lixia Zhang[1]

[1] University of California, Los Angeles
[2] ThousandEyes
[3] Level-3 Communications Inc.
[4] Arbor Networks

**Abstract.** BGP is a hard-state protocol that uses TCP connections to reliably exchange routing state updates between neighbor BGP routers. According to the protocol, only routing changes should trigger a BGP router to generate updates; updates that do not express any routing changes are superfluous and should not occur. Nonetheless, such 'duplicate' BGP updates have been observed in reports as early as 1998 and as recently as 2007. To date, no quantitative measurement has been conducted on how many of these duplicates get sent, who is sending them, when they are observed, what impact they have on the global health of the Internet, or why these 'duplicate' updates are even being generated. In this paper, we address all of the above through a systematic assessment on the BGP duplicate updates. We first show that duplicates can have a negative impact on router processing loads; routers can receive upto 86.42% duplicates during their busiest times. We then reveal that there is a significant number of duplicates on the Internet - about 13% of all BGP routing updates are duplicates. Finally, through a detailed investigation of duplicate properties, we manage to discover the major cause behind the generation of pathological duplicate BGP updates.

## 1 Introduction

BGP is the de facto standard inter-domain routing protocol used to exchange destination reachability information on the Internet. BGP was designed as a hard-state protocol, so all BGP updates sent by a router should always communicate some change or addition to the most current routing information reported by the router [7]. However, actual observations of BGP dynamics reveal that routers tend to occassionally send BGP updates with absolutely no change to the most current routing information reported by the router. In fact, there are many cases where routers send exact copies of the most recent update previously sent. To date, there has been no explanation as to why these 'duplicate' routing updates occur in BGP today.

Existence of duplicate updates in BGP was first reported in 1998. Labovitz's [2] seminal work on BGP measurements showed that the actual number of BGP updates observed were an order of a magnitude more than expected. Labovitz revealed that a large portion of the total updates were in fact duplicates, and he attributed this to problems with routers from specific vendors. The industry quickly responded with a software fix to address the duplicate generation problem, and it was believed that the fix would eliminate the duplicate pathology observed in [2]. However, in 2007 Li *et al.* [4] re-examined the health of BGP dynamics and discovered that, despite industry attempts to stop duplicate generation, duplicates were still seen in BGP. To date, nobody has been able to determine the cause of these duplicates. There also have never been any reports on the effects, if any, that duplicates have on Internet health.

In this paper, we make the following contributions.

- We provide a better understanding of the nature of duplicate generation by quantifying the amount of duplicate updates from different points on the Internet. We also look at duplicates from different moments in time.
- We reveal the impact of duplicates on Internet health. Unlike the common belief that duplicates are relatively benign, we show that they can negatively impact the instantaneous router processing load.
- As part of our work towards understanding duplicates, we provide a methodology for mapping eBGP updates to their corresponding iBGP updates. We believe that our methodology can be useful toward future studies that require a mapping of eBGP to iBGP updates, or vice versa.
- Using our observations of duplicate behavior, we manage to finally determine the exact cause behind duplicate generation.

## 2 Background

In this section, we review some routing details that are particularly relevant to our study of duplicates. Specifically, we discuss the definition of ‘duplicate updates’ and BGP peering topologies.

### 2.1 Definition of Duplicates

A BGP update for prefix $p$ sent by router $r$ is a ‘duplicate’ if and only if all attributes in the update are the same as the most recent previous update for prefix $p$ sent by router $r$, and both the update and the previous update belong to the same BGP session.

### 2.2 BGP Peering Topologies

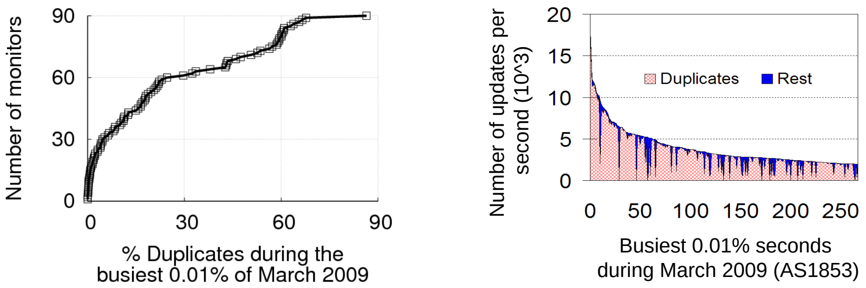
Today, BGP is used for both inter-domain routing (eBGP), as well as intra-domain routing (iBGP). Here we briefly describe the common peering topologies for both inter and intra-domain routing.

**External BGP:** When BGP is used to convey reachability information between two routers that reside in different domains (inter-domain routing), the session between these two routers is called an eBGP session. The routing information in each update is conveyed in the form of BGP attributes. Some of the more relevant attributes to this paper are Next-hop, MED, Local-pref, and Community.

**Internal BGP:** iBGP is used to distribute reachability information received from eBGP peers to routers within one domain. To avoid forming a routing information loop, it was originally required that all iBGP speakers are fully meshed and the reachability learned from an iBGP speaker is not propagated to another iBGP speaker. In practice, this approach is not scalable and too expensive to manage. This leads to the use of route reflection (RR) [1] and AS confederations [8], which relaxes this full-mesh requirement among all iBGP peers. However, having to traverse more than one hop for an update from an iBGP peer to another iBGP peer re-introduces the possibility of routing information loops under both schemes. To avoid forming a routing loop, route reflection and AS confederation define new attributes, namely Cluster-list and AS-confed-sequence respectively, and use them in the similar way that AS-path is used in eBGP.

**iBGP and eBGP interaction:** A router that peers with both iBGP peers and eBGP peers changes or even removes certain attributes when it sends reachability information received from an iBGP peer to its eBGP peer. Some attributes defined both in iBGP and eBGP such as Next-hop, MED, and Local-pref may have changes in their values before sent out to eBGP peers. Furthermore, certain attributes that are only defined in iBGP such as Cluster-list and AS-confed-sequence are removed and not sent out to eBGP peers.

## 3 Impact of Duplicates on Routers



(a) Distribution of % duplicates

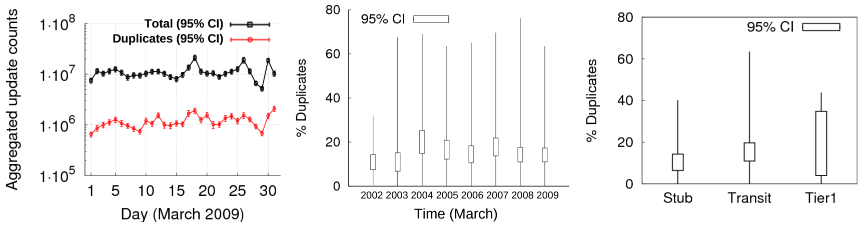
(b) # of duplicates during the busiest seconds

**Fig. 1.** Impact on processing loads

We start by measuring the impact that duplicates have on Internet health. Up until now, it was believed that duplicates do not hinder routing efficiency in any significant way [3]. However, we find that duplicates are responsible for the majority of router processing loads during their busiest times. Previous studies have shown that higher processing loads can lead to more session resets, routing loops, and packet losses [9]. Thus, we measured how much duplicates contribute to the router processing loads during their busiest times during the month of March 2009. We define 'busiest times' as the top 0.01% of seconds within which the largest number of updates were generated. Our data set consists of a specific subset of all RouteViews/RIPE monitors. The monitors were carefully chosen such that each monitor was available for the entire month of

| Year | 2002 | 2003 | 2004 | 2005 | 2006 | 2007 | 2008 | 2009 |
|---|---|---|---|---|---|---|---|---|
| Number of selected monitors | 27 | 37 | 54 | 67 | 79 | 100 | 109 | 90 |
| Number of total updates ($10^6$) | 129.5 | 207.3 | 316.4 | 426.5 | 423.7 | 511.2 | 652.2 | 677.4 |
| Number of duplicate updates ($10^6$) | 12.7 | 32.0 | 68.9 | 74.6 | 63.8 | 137.1 | 111.0 | 91.3 |

**Table 1.** Aggregated number of updates and duplicates



(a) During March 2009 (b) From 2002 to 2009 (c) From different AS type

**Fig. 2.** Amount of duplicate updates

March 2009 and that there was at most one monitor per AS in our dataset. The number of stub, transit, and tier-1 monitors we ended up with were 27, 55, and 8 respectively, for a total of 90 monitors. We preprocess all of our data using the minimum collection algorithm (MCT) [10] to filter out updates due to session resets before performing any of the measurements presented in this paper.

Figure 1(a) shows the percent of duplicate traffic during busiest times for each of the 90 ASes we monitored. Notice that for 22% (20 out of 90) of all monitored ASes, duplicates contribute 50% or more of the update traffic during busiest times. Later in Section 5, we describe how these duplicate bursts are generated in detail as we reveal the causes of duplicates. Figure 1(b) is a close-up look at a particularly bad case of our measurement, AS1853. Overall, 86.42% of total updates during the top 0.01% of busiest times were duplicates. During the busiest second the router in AS1853 had to process about 175,000 updates in Figure 1(b).

## 4 Understanding Duplicates across Time and Space

Now that we understand the negative impact that duplicates can have on Internet health, we analyze duplicate generation in detail to gain a better understanding of this duplicate pathology, and maybe even discover the cause of duplicate generation. Not only do we measure the prevalence of duplicates updates on the Internet today, we also measure the number of duplicates that we have seen over the past few years. We then explore whether topological factors (such as size of AS or connectivity) show any correlation with occurrences of duplicates. Our data set consisted of the same 90 monitors we used for our measurements in section 3.

### 4.1 Are Duplicates Observed at All Times?

Figure 2(a) shows the amount of duplicates along with the total number of updates from all 90 monitors during March 2009. It turns out that duplicate generation is not just a pathological behavior rarely seen on the Internet. In this month alone, the total aggregated number of updates was about 677 million. Among those, about 91 million updates were duplicates. Thus duplicates make up 13.4% of aggregated BGP traffic.

Figure 2(b) shows how long duplicates have existed in BGP by showing the maximum, minimum, and 95% confidence intervals of % duplicates observed by different monitors for the month of March from 2002 through 2009. For each year, we selected monitors based on the criteria described in section 3. Table 1 shows the number of monitors we used from 2002 through 2009. The number of qualified monitors generally increase over time, mainly because more ASes peered with RouteViews and RIPE over time.[5] We performed the same measurement for other months from 2002 through 2009, and the results were all similar. The amount of duplicates we counted also agree with the amount observed in previous studies [4].

### 4.2 Are Duplicates Observed from All Networks?

Our next measurement is aimed at understanding if size or type (e.g. stub, tier-1) of network has any correlation with observed duplicates. We measured the percentage of duplicates out of total updates that each network generated for the month of March 2009.

Figure 2(c) summarizes our findings. All three types of networks generate duplicates with some variation in their percentages. The large confidence interval range for tier-1s is mainly due to the small number of data points available to us. Minimum % duplicates were very low in all three cases. At the same time maximum % duplicates were quite high for all types, showing a large variation in behavior even amongst networks of the same type. Later in section 6, we discuss why the amount of duplicates observed varies so widely amongst networks of the same type.

### 4.3 Where Do Duplicates Originate?

So far, we have observed duplicates from different monitors. However, we do not quite know where these duplicates originate. By specification, a BGP router should not propagate a duplicate it receives. Thus, when we observe a duplicate at AS X with a path X-Y-Z, where Z is the origin AS, we hypothesized that the duplicate message must be generated by X and not by Y or Z. Our next exercise is to verify our hypothesis.

For this, we looked specifically at duplicates for particular prefixes where the following was true. First, the observed duplicate for prefix $p$ from AS X had an AS-path ending with X-Y. Second, we had to have monitors for both AS X and AS Y. With this, we can see whether the duplicates actually originate at (or within) AS X, or whether

[5] The exception was between 2008 and 2009. This was because some of the collectors in RIPE had problems during March 2009, and we did not use any monitors that did not have complete data for the month.

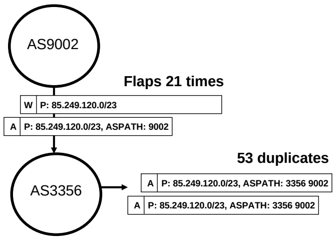

AS9002
Flaps 21 times
W
P: 85.249.120.0/23
A
P: 85.249.120.0/23, ASPATH: 9002
53 duplicates
AS3356
A
P: 85.249.120.0/23, ASPATH: 3356 9002
A
P: 85.249.120.0/23, ASPATH: 3356 9002

**Fig. 3.** External view of duplicate generation

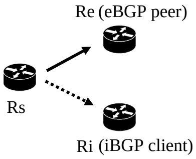


**Fig. 4.** Data collection

they were sent to X from Y. Our case study consisted of prefix 85.249.120.0/23 advertised by AS 9002, a direct customer of AS 3356. We had monitors in both AS 9002 and AS 3356.

Figure 3 summaries our results. During March 2009, AS 9002 announced and withdrew prefix 85.249.120.0/23 21 times. Upon receipt of these announcement and withdrawal pairs, AS 3356 sends out the announcement to the monitor with prepended AS-path, but AS 3356 never sends the withdrawal. Instead, AS 3356 sends a duplicate announcement to our monitor. In total, AS 3356 generates 53 duplicates on prefix 85.249.120.0/23 after receiving 21 pairs of announcement and withdraw messages. Not only does this observation back up our hypothesis that the sender of duplicates is the originator of duplicates, but it also suggests that the cause of duplicates may have something to do with the way internal topology dynamics interact with eBGP updates.

## 5 Discovering the Cause of Duplicates

Once we suspected that duplicates may be generated due to some interaction between iBGP and eBGP, we ran an experiment designed to compare eBGP update+duplicate pairs, match them with their iBGP counterparts, and compare these iBGP updates to see what we might learn about duplicate generation.

### 5.1 Passive Measurement using iBGP and eBGP Data

Our first step was to obtain the data needed for our investigation. We teamed up with a tier-1 ISP who provided us with access to both iBGP and eBGP updates generated by one of their routers. Figure 4 illustrates our data-collection setup. $R_s$ is the router sending updates to our two collector boxes, $R_i$ and $R_e$. $R_i$ is configured as an iBGP client of $R_s$ (*i.e.* route reflector client), collecting iBGP data from $R_s$. $R_e$ is an eBGP peer of $R_s$, collecting eBGP updates from $R_s$. Both the iBGP and eBGP sessions have their MRAI timers disabled, so that $R_s$ will send updates to our collectors as $R_s$ has updates to send.

Now that we obtained the necessary data, we needed a way to match up eBGP updates to their corresponding iBGP updates for comparison. There are two challenges in mapping iBGP update sequence with that of eBGP. First, the time that two updates, triggered by the same event, are sent out from $R_s$ can be different. This is due to the non-deterministic nature of $R_s$. Second, $R_i$ and $R_e$'s system clocks may not be synchronized. We resolve these timing issues by introducing the notion of update 'signatures', which we now describe.

$$sig(u) = \text{peer} \parallel \text{asn} \parallel \text{prefix} \parallel \text{aspath} \parallel \text{origin} \parallel \text{comm} \parallel \text{agg}$$

The signature of an update contains all of BGP's transitive attributes that should be the same in $R_s$'s updates to either $R_i$ or $R_e$. By using the notion of signature, we calculate the time differences $t_d$ observed between eBGP updates and their iBGP counterparts. We first generate signatures of all updates received during the $t_{ebgp}$ second, and then search for the second in iBGP, $t_{ibgp}$, that yields the maximum fraction of matched signatures. In our case, the peak fraction of matched signatures was about 0.7 at a lag value of 0 (*i.e.* $t_d$ = 0). The remaining 0.3 were dispersed within a 10-second range centered at $t_{ebgp}$. This means that the system times of $R_i$ and $R_e$ have synchronized system clocks to the second precision.

After discovering $t_d$, we were able to map eBGP updates to their iBGP counterparts using a heuristic algorithm involving signature and timestamp comparisons. We collected one day of iBGP and eBGP updates, putting them in sequential order as sent from $R_s$. We start with the first eBGP update in the sequence. As we moved down the sequence, we kept per-prefix history of signatures for every update we encounter for a time window of 60 seconds. For each eBGP duplicate update for prefix $p$ we found as we moved forward, we looked at the corresponding iBGP time window to find a match for the sequence of signatures we recorded in eBGP for this prefix $p$. We say the sequence has a match when there is the *exact* sequence of update signatures within the iBGP time window. Using our heuristic, we were able to match 95.61% of eBGP update+duplicate pairs to their iBGP counterparts. 4.39% of eBGP update+duplicate pairs could not be mapped to any iBGP counterparts. The missing pairs were due to how the router processed updates. [6]

After mapping eBGP updates to their iBGP counterparts, we took each eBGP update+duplicate pair and compared the contents of their corresponding iBGP updates. For 100% of the 176,266 matched ebgp+duplicate pairs, we observed that their iBGP counterparts had differing non-mandatory attribute values. Table 5 shows our results. 0.15% of pairs were exceptions, only differing in MED values. For the other 99.85% of eBGP update+duplicate pairs, we observed corresponding iBGP update pairs with either Cluster-list and/or Originator-id differences. These attribute differences represent changes in intra-domain routing path selections.

### 5.2 The Cause of Duplicates

The results of our experiment allowed us to determine the main cause of eBGP duplicate updates. Our theory proved to be correct; duplicates are caused by an *unintended interaction* between eBGP and iBGP. The reason that duplicates are generated is that

[6] When two or more updates are received on the same prefix in a very short time, the router sometimes sends out different number of updates to different peers. So, there were cases that the number of updates sent to iBGP client is different than that of updates sent to eBGP peer, in which case we declared that there is no match.

| eBGP duplicate count | % Total | Observed iBGP differences |
|---|---|---|
| 173,594 | 94.77 | Cluster-list only |
| 244 | 0.13 | Cluster-list and others |
| 1,371 | 0.75 | Originator-id and others |
| 1,057 | 0.58 | Cluster-list + Originator-id + others |
| 269 | 0.15 | MED |
| 6,647 | 3.63 | No match found |
| Total: 183,182 | 100.00 | |

**Fig. 5.** Matched iBGP updates

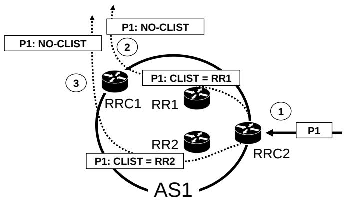


**Fig. 6.** Inferred cause of duplicates

routers are receiving updates via iBGP which differ in iBGP attribute values alone, and thus the router believes the updates to be unique. However, once the router processes the update, strips the iBGP attribute values, and sends the update to its eBGP peer, the two updates look identical from the point of view of the eBGP peer. [7] Figure 6 illustrates a case where duplicates are generated due to changes in an iBGP attribute (Cluster-list in this case).

The main cause of eBGP duplicate updates showed that certain iBGP attribute changes (Cluster-list and Originator-id) can generate eBGP duplicate updates. We wondered if other iBGP attribute changes might also generate eBGP duplicate updates. To check for this, we performed a simple controlled experiment. We set up two ASes (AS1 and AS2). In AS1, we placed a BGP update injector and a router $R_1$. The injector maintains an iBGP session with $R_1$ and sends controlled iBGP updates. $R_1$ peers with a router, $R_2$, in AS2 using an eBGP session. [8]

After injecting pairs of iBGP updates that only differ in one attribute, we observed that a pair of iBGP updates differing in either Next-hop, Local-pref, or MED attributes will generate an eBGP duplicate update.

The experiments we have done so far shed light on how the duplicate bursts, which we discussed in Section 3, are generated. When a router used to reach a set of prefixes fails, this failure (or flapping) event generates updates that only differ in Next-hop for the set of prefixes. All of these updates become duplicates as they are sent to the eBGP peers. Using the iBGP/eBGP data collected from our tier-1 ISP, we verified that indeed duplicate bursts are preceded by an iBGP route flapping.

## 6 Differences in the Amount of Observed Duplicates

As observed in 4.2, ASes of the same type vary in the proportion of duplicates they generate. One reason may be a difference in MRAI timer settings amongst the networks.

[7] In our study, we observed that duplicates are generated due to changes in Cluster-list and Originator-id oscillations under route reflection. In a similar way, we believe ASes using AS confederation architectures will also generate duplicates due to the use of a non-mandatory non-transitive attribute named AS-confed-sequence, which is essentially the AS confederation version of the Cluster-list attribute under route reflector architectures.

[8] Here, $R_1$ is a Cisco 7200 router running IOS v12.2, and $R_2$ is a Quagga router which we use as a BGP update collector.

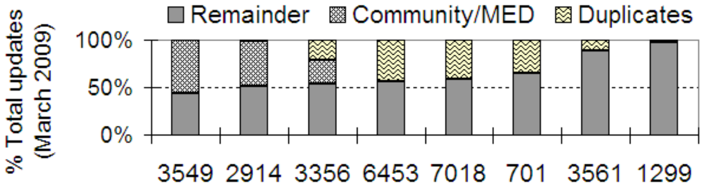


**Fig. 7.** Other potential noises

Duplicates are generated during internal routing changes. During the changes, updates come in bursts, and thus MRAI timers can prevent many updates from being sent.

MRAI timer differences do not fully explain why the amount of observed duplicates varies so much from one AS to another. During our experiments involving eBGP and iBGP interactions, we noticed that Cluster-list changes were often coupled with a change in Community or MED attribute values. In these cases, we observed potentially wasteful updates with fluctuating Community/MED values rather than duplicates. We asked operators at our tier-1 ISP and they confirmed that this was quite deliberate; routers were configured to make changes in certain transitive attribute values whenever there was a change in certain non-mandatory attribute values in accordance with [5, 6]. [5, 6] suggests using Community attribute values as a general purpose attribute to convey informational tags as well as action tags to receiving networks. MED values were also used for traffic engineering purposes. However, operators admit that not all peers need or use this Community information, and for those routers that do not use the Community information, these BGP updates are as useless to them as duplicates. However, such updates can be more detrimental than duplicates in one significant way; with duplicates, the negative impact is limited to the direct neighbors. As described earlier, duplicates do not travel more than one hop. However, if some other (optional) transitive attributes such as Community is changed, then the update is no longer a duplicate and can potentially be propagated more than one hop.Community value changes are not useful to networks that are more than one hop away, and yet these networks still must suffer the same negative impacts of receiving a superfluous BGP update.[9]

Our discovery of these potentially wasteful BGP updates led us to wonder if other ASes generated similarly potentially wasteful updates. We looked at all updates from tier-1s observed by our monitors for the month of March 2009, and classified the updates into 3 types - duplicates, Community/MED change, and remainder. Figure 7 shows our results. While AS3549 and AS2914 generated almost no duplicates, 50% or more of their total updates were Community/MED change updates. We suspect that many of these updates could be useless to many networks that receive the update. We intend on verifying our suspicion in future work.

[9] Different router vendors implemented different default behavior in sending Community attribute. For ISPs that use network equipment where the default behavior is to send communities (*e.g.* Juniper), then the effect of this problem are likely to be amplified. However, for ISPs that use network equipment where the default behavior is to not send communities by default (*e.g.* Cisco), then the effect of this problem are likely to be less.

## 7 Conclusion

In this paper, we conducted the first comprehensive measurement study quantifying the prevalence of duplicates on the Internet across space and time. We discovered that duplicates make up over 10% of all BGP update traffic. We examined the impact that duplicates have on the overall health of the Internet, and discovered that routers can receive upto 86.4% duplicates during their busiest times. We developed a heuristic to match eBGP updates with their corresponding iBGP counterparts. Finally, we combined our observations with our heuristic to discover the causes of duplicates on the Internet - duplicates are caused by an *unintended interaction* between iBGP and eBGP.

While pure duplicates are clearly unnecessary BGP overhead, our work revealed that duplicates may not be the only superfluous BGP updates floating around on the Internet. As described in section 6, updates that couple non-transitive attribute changes with transitive attribute changes may not be useful to all recipients. It would be interesting to identify all forms of superfluous BGP updates and gain an exact measure of how much BGP traffic is simply unwanted noise. We hope that our work allows the Internet community to take a significant step towards a optimal and clean routing communication system.

## References

1. T. Bates, E. Chen, and R. Chandra. RFC 4456: BGP Route Reflection: An Alternative to Full Mesh Internal BGP, April 2006.
2. C. Labovitz, G. R. Malan, and F. Jahanian. Internet routing instability. *ACM/IEEE Transactions on Networking*, 6(5):515–528, October 1998.
3. C. Labovitz, G. R. Malan, and F. Jahanian. Origins of internet routing instability. *ACM/IEEE Infocom*, 1:218–226, March 1999.
4. J. Li, M. Guidero, Z. Wu, E. Purpus, and T. Ehrenkranz. BGP Dynamics Revisited. In *ACM Sigcomm Computer Communications Review*, April 2007.
5. D. Meyer. RFC 4384: BGP Communities for data collection, 2006.
6. R. A. Steenbergen and T. Scholl. BGP Communities: a guide for service provider networks, 2007.
7. P. Traina. RFC 1774: BGP-4 protocol analysis, 1995.
8. P. Traina, D. McPherson, and J. Scudder. RFC 5065: Autonomous System Confederations for BGP, August 2007.
9. L. Wang, X. Zhao, D. Pei, R. Bush, D. Massey, A. Mankin, S. F. Wu, and L. Zhang. Observation and analysis of BGP behavior under stress. In *IMW '02: Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurment*, pages 183–195, New York, NY, USA, 2002. ACM.
10. B. Zhang, V. Kambhampati, M. Lad, D. Massey, and L. Zhang. Identifying BGP routing table transfers. In *MineNet '05: Proceedings of the 2005 ACM SIGCOMM workshop on Mining network data*, pages 213–218, New York, NY, USA, 2005. ACM.