The Origin of BGP Duplicates

D. Hauweele[1†] B. Quoitin[1] C. Pelsser[2‡] R. Bush[3]

[1] *Computer Science Dept., University of Mons (UMONS), Belgium*
[2] *University of Strasbourg & CNRS, France*
[3] *Internet Initiative Japan (IIJ)*

The Border Gateway Protocol propagates routing information accross the Internet in an incremental manner. It only advertises to its peers changes in routing. However, as early as 1998, observations have been made of BGP announcing the same route multiple times, causing router CPU load, memory usage and convergence time higher than expected. In this paper, by performing controlled experiments, we pinpoint multiple causes of duplicates, ranging from the lack of full RIB-Outs to the discrete processing of update messages.

Keywords: BGP, duplicates, Internet routing, network measurement

1 Introduction

The Border Gateway Protocol [RLH06] (BGP) is the de facto standard used to exchange inter-AS routing information on the Internet. This information is exchanged by the mean of update messages which notify the reachability or non-reachability of network prefixes. According to the protocol specification, a BGP speaker should not issue an update containing the same BGP information as was most recently advertised for the prefix. However such redundant messages, called *duplicate updates*, have been observed as early as 1998 [LMJ98]. The authors noted that the resulting high level of instability was detrimental to the operations of the Internet, causing high router CPU load, making routers unresponsive and in the worst cases leading to packet or routing information losses. In addition, they may sometimes trigger unreachability when interacting with route flap damping [PMM+11].

Several studies later revisited BGP dynamics [LABJ00, LGW+07, PJL+10, EKD12, ED13] and its impact on router CPU load [ACBD04], some focused on BGP duplicates. Although the number of pathological updates declined over time, duplicates still constitute a significant part of the BGP traffic with up to 15% of the updates observed at RIPE monitors in 2006 [LGW+07]. It was later shown that the duplicate problem is even worse for routers in the core of the Internet with the portion of duplicates varying from 7% to 60% in 2008 [EKD12]. More recently, in 2009, Park et al. [PJL+10] studied over 90 RouteViews/RIPE monitors and showed that the duplicates make up 13.5% of the aggregated BGP traffic. Routers can receive up to 86.4% of duplicates during their busiest time. These previous works show that duplicates are a continuing problem. We confirm this observation by looking at all sessions from EQUINIX, ISC, LINX and WIDE RouteViews collectors from 2009 to 2014. 48.5% of the traces we observed had more than 10% of duplicates. The traces also display a high variability with an average of $(18.84 \pm 22.31)\%$ duplicates over the whole period and $(23.16 \pm 25.73)\%$ in 2014. Confidence interval are within one standard deviation. Finally, [PJL+10] hinted that a change in attributes attached to iBGP routes may trigger eBGP duplicates. To the best of our knowledge, so far no thorough study has explained their origin.

In Section 2 we discuss the causes of today's duplicates. Although the majority of duplicates in 1998 were bogus route withdrawals, this is not the case today (less than 0.5% on almost all traces). To understand what causes duplicates, we inject carefully crafted BGP updates into a router and we correlate the input and output BGP traffic. Based on this, we identify different causes for duplicates.

[†] David started this work during his internship at IIJ.
[‡] The credits go to IIJ for supporting Cristel's work.

2 The origin of duplicates

To investigate the origin of BGP duplicates, we follow two different approaches. First we identify cases of duplicates in live BGP feeds. Second we perform a fully controlled experiment where we inject crafted sequences of messages into a test router. Our experiment allows to confirm the hypotheses of Park et al. on the origin of duplicates. We also go much further as we establish additional causes for duplicates.

2.1 Definitions

We define a duplicate as a *redundant* prefix advertisement with the *same attributes* as the most recent update for this prefix on the same session and not interleaved with a withdrawal or a session reset. This definition is stricter than the one in [LMJ98] where an update is considered a duplicate (AADup) if its AS-Path and next-hop do not change.

We also define the *ratio of duplicates* as the number of duplicates (including the original messages) over the total number of messages. With this definition, a trace where every route advertisement is duplicated will have a ratio of 100%.

2.2 Real BGP feed experiment

We describe in the following paragraphs a common case of duplicates that we observed in a live BGP feed. Other cases were also observed but are not presented due to space constraints. Our setup is shown in Fig. 1. Devices $r0$, $r1$ (Cisco) and $r2$ (Juniper) are real routers while $mon0$ is a dedicated host running a software BGP router (Quagga). The router under test is $r2$. It receives BGP messages from $r0$ and $r1$ through *input* eBGP sessions. After selecting its best routes, $r2$ sends BGP messages over a single *output* eBGP session to $mon0$. The routes learned by $r0$ and $r1$ are from real BGP feeds received in September 2013 for a duration of 23 days. The $mon0$ host captures all the BGP messages received on the *mirror* and *output* sessions. The *mirror* sessions (dashed lines on Fig. 1) allow to capture the *input* routes advertised by the upstream routers $r0$ and $r1$. To reduce timing differences between the *input* and *mirror* sessions, both sessions are placed in the same update group on $r0$ and $r1$. The MRAI is also set to zero on these routers.

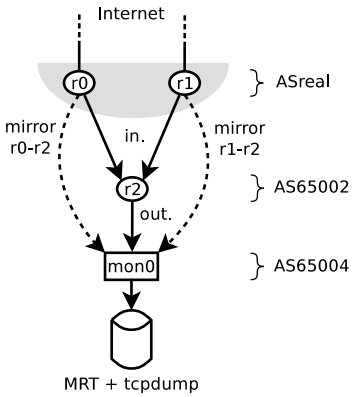

Figure 1: Setup for the real BGP feed experiment.

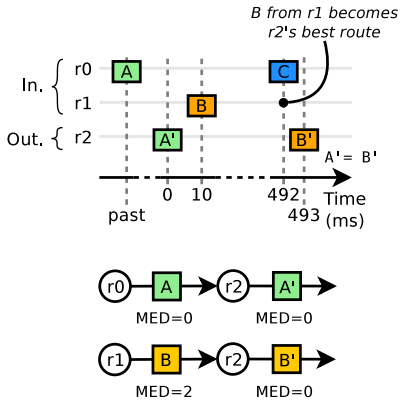

Figure 2: The MED case.

Fig. 2 illustrates the MED case. Three different *input* routes are involved, all for the same IPv4 prefix. The first route, A, has an AS Path of length 5 and a MED value of 0. The second route, B, has the same AS Path as A but a MED value of 2. The third route, C, has an AS Path of length 6 and a MED value of 0. At time 0ms, $r2$ announces route A learned from $r0$. Before announcing A, $r2$ updates the AS-Path and strips the MED, which produces route A'. At time 10ms, $r1$ announces route B to $r2$. The decision process of $r2$ ranks route A better than route B, causing no change in $r2$'s best route. At time 492ms, $r0$ announces to $r2$ route C which has a longer AS Path. Route C implicitly withdraws route A. As a consequence, $r2$ now selects route B as best. Before announcing B, $r2$ strips the MED value, producing B'. *Output* routes A' and B' are equal, hence B' is a duplicate of A'.

We believe the duplicate in this MED case is due to the MED attribute being stripped at the output of $r2$, suggesting that it does not fully implement a stateful BGP.

2.3 Controlled experiment

To confirm the hypotheses of the previous section and to extend those from previous work [PJL+10], we perform the same input/output matching in a fully controlled experiment. We systematically test an exhaustive set of situations that may not have appeared in the setting with a real live BGP feed. We are able to find additional causes of duplicates and pinpoint more precisely the reasons behind these duplicates.

Table 1 summarizes the results of the injection experiment. Due to space limitations, only results for a small number of test cases are presented. For each experiment, the first column shows the average delay between messages observed on the *input* and its standard deviation. The second column shows the same information for the *output*. The last column shows the ratio of duplicates, that is, the number of duplicates including the initial update over the number of updates (see Section 2.1).

Test case	**Input (ms)**	**Output (ms)**	**Dup.**
`NotVisible`	–	–	100%
`RFlap` (1 ms)	1.23 ± 0.50	3.47 ± 3.46	69.0%
`RFlap` (2 ms)	2.07 ± 0.39	2.84 ± 0.99	25.9%
`RFlap` (3 ms)	3.07 ± 0.44	3.06 ± 0.48	0.1%

Table 1: Results of selected injection test cases.

2.3.1 Internal / non-transitive / filtered attributes

This first set of experiments (`NotVisible`) considers the case of attributes whose changes should not be visible from the outside of an AS as they are either internal, non-transitive or filtered/rewritten by output policies. The objective of these experiments is to test whether or not such attributes could cause duplicate routes to be sent by the router.

For this purpose, we repeatedly send a sequence of 2 route updates (A,B) for the same destination prefix. Route B differs from route A by only a specific internal / non-transitive / filtered attribute. The expected behavior is as follows. When route A is received, it is selected as best as there is no other choice. It is then propagated on the output session. When route B is received, it replaces route A (implicit withdraw). Route B should not be propagated to the *output* session as it differs from route A only by an attribute that is either internal, non-transitive, or removed by a filter. Hence, on the *output* session, routes A and B are identical.

We observe a duplicate ratio of 100% for experiments in this class independently of the updates inter-arrival times, as shown in Table 1 for the test case `NotVisible`. The router was not able to detect that the second route was a duplicate of the previous one. We explain this behavior from the statelessness of the BGP implementation.

These results held for the following attributes: MED, Local Pref, Cluster List, and Originator ID. We also observed a 100% duplicates ratio for non-transitive Community values stripped by outgoing policies and for rewritten Next-Hop.

2.3.2 Fast flapping route

In a second set of experiments (`RFlap`) we investigate the impact of a flapping route on the generation of duplicates. The experiment relies on the repetition of a simple sequence of 2 BGP updates (A,W) for the same prefix. A announces a route while W withdraws it. We observed similar results for a transitive attribute that flaps from one value to another and back.

The objective of this experiment is to trigger duplicates by forcing a route to change multiple times before the router has the opportunity to propagate it. To understand this behavior, we need to refine our model of how a router generates updates. When a route towards a prefix changes, the main BGP process does not send an update immediately. Instead, this task is delegated to a separate thread that periodically reads the RIB and advertises the routes marked as changed.

The following scenario illustrates how the transmission of a duplicate update can be caused. When the first Announce is received, the route is marked as changed in the RIB. The RIB is then scanned and an update is sent. Then, the Withdraw is received and the route is again marked as changed. However, before the RIB is scanned, the third message (second Announce) is received and the route is again marked as changed. When the RIB is scanned, the second Announce, identical to the first one is sent. It is a duplicate as the router did not have time to send a Withdraw between the two Announces.

We repeat this experiment with increasing delay between updates: 1ms, 2ms and 3ms. The results are in Table 1 for test case `RFlap`. We observe that with a 1ms interval, almost 70% of output updates are duplicates. When the interval between input updates increases, the ratio of duplicates decreases. With a 2ms interval, the ratio is almost 26% and at 3ms, there are almost no duplicates.

We also tested the impact of the MRAI on the generation of duplicates. We conducted the same experiment with a larger interval of 2 seconds and a MRAI set to 6 seconds. With this experiment we generated more than 30% of duplicates.

3 Conclusion

Redundant consecutive BGP announcements consume unnecessary bandwidth and CPU in routers. In addition, these messages delay the propagation of useful routing information. We measured that these duplicates still constitute a large part of today's BGP traffic with an average of 23.16% duplicates in 2014.

In order to identify specific cases of duplicates, we first looked at a router that receives live BGP feeds. This is an approach similar to previous studies that gave us some initial insight on potential causes for duplicates. Then we injected synthetic updates into a real router and observed the presence of duplicates at its output. Our experiments allowed to identify two main causes of duplicates that can be attributed to the statelessness and discrete nature of BGP implementations: changes in attributes that are not propagated further and flapping of routes or attributes. Finally, we also observed that the current implementation can generate duplicates when sets are not considered equal if ordered differently.

References

[ACBD04] S. Agarwal, C.-N. Chuah, S. Bhattacharyya, and C. Diot. Impact of BGP dynamics on router CPU utilization. In *Passive and Active Network Measurement*, pages 278–288, 2004.

[ED13] A. Elmokashfi and A. Dhamdhere. Revisiting BGP churn growth. *SIGCOMM CCR*, 44(1):5–12, December 2013.

[EKD12] A. Elmokashfi, A. Kvalbein, and C. Dovrolis. BGP churn evolution: a perspective from the core. *Networking, IEEE/ACM Transactions on*, 20(2):571–584, 2012.

[LABJ00] C. Labovitz, A. Ahuja, A. Bose, and F. Jahanian. Delayed internet routing convergence. *ACM SIGCOMM CCR*, 30(4):175–187, 2000.

[LGW+07] J. Li, M. Guidero, Z. Wu, E. Purpus, and T. Ehrenkranz. BGP routing dynamics revisited. *ACM SIGCOMM CCR*, 37(2):5–16, 2007.

[LMJ98] C. Labovitz, G. R. Malan, and F. Jahanian. Internet routing instability. *Networking, IEEE/ACM Transactions on*, 6(5):515–528, 1998.

[PJL+10] J. H. Park, D. Jen, M. Lad, S. Amante, D. McPherson, and L. Zhang. Investigating occurrence of duplicate updates in BGP announcements. In *PAM*, pages 11–20, 2010.

[PMM+11] C. Pelsser, O. Maennel, P. Mohapatra, R. Bush, and K. Patel. Route flap damping made usable. In *PAM*, pages 143–152, 2011.

[RLH06] Y. Rekhter, T. Li, and S. Hares. A Border Gateway Protocol 4 (BGP-4). RFC 4271 (Draft Standard), January 2006. Updated by RFCs 6286, 6608, 6793.