Preventing the Unnecessary Propagation of BGP Withdraws *

Virginie Van den Schrieck[1], Pierre Francois[1], Cristel Pelsser[2], and Olivier Bonaventure[1]

[1] Universite catholique de Louvain (UCL), CSE Dept
Place Sainte-Barbe 2
1348 Louvain-la-Neuve, Belgium
firstname.lastname@uclouvain.be
[2] NTT Corporation, NTT Network Service Systems Laboratories
9-11, Midori-Cho 3 Chrome
Musashino-shi, Tokyo 180-8585, Japan
pelsser.cristel@lab.ntt.co.jp

Abstract. Due to the way BGP paths are distributed over iBGP sessions inside an Autonomous System (AS), a BGP withdraw that follows a failure may be propagated outside the AS although other routers of the AS know a valid alternate path. This causes transient losses of connectivity and contributes to the propagation of a large number of unnecessary BGP messages. In this paper, we show, based on RouteViews data, that a significant number of BGP withdraws are propagated even though alternate paths exists in another border router of the same AS. We propose an incrementally deployable solution based on BGP communities that allows the BGP routers of an AS to suspend the propagation of BGP withdraws when an alternate path is available at the borders of their AS.

Key words: BGP, Internet, Churn

1 Introduction

The Border Gateway Protocol (BGP) [1] plays a key role in today's Internet as it allows Internet Service Providers (ISPs) and enterprise networks to announce routes towards their IP prefixes. During the last years, network operators and researchers have been concerned by the limits to the scalability of the Internet architecture and BGP in particular [2]. An important problem that affects BGP is the BGP churn, i.e. the number of BGP messages exchanged among BGP routers.

BGP is a path vector protocol with two different types of BGP messages : updates and withdraws. A BGP update is used to advertise a path towards a prefix or a change

* The research results presented herein have received support from Trilogy (http://www.trilogy-project.eu), a research project (ICT-216372) partially funded by the European Community under its Seventh Framework Programme. The views expressed here are those of the author(s) only. The European Commission is not liable for any use that may be made of the information in this document.

in a previously announced path towards a prefix. A BGP withdraw indicates that a previously announced prefix becomes unreachable. BGP routers exchange BGP messages over a BGP session. An analysis of the BGP messages exchanged in the global Internet shows that their number is very high [2]. This BGP churn causes high-CPU load on smaller BGP routers. Several causes have been identified. First, some interdomain links are unstable and fail frequently [3–5]. Each of these failures causes the transmission of a number of BGP withdraws. Second, as BGP relies on path vectors, it suffers from the path exploration problem when a route becomes unavailable[6]. When a route fails, a new BGP convergence starts. During this convergence, routers may advertise paths that they consider valid although they are also affected by the failure. These paths will be withdrawn later causing another exchange of BGP messages. The MRAI timer [1] and the route flap damping mechanism [7] may further delay this convergence. Third, due to their routing policies and internal BGP organization [8], some BGP routers from large ASes may transiently send BGP withdraws although alternate paths are available inside the AS, because those alternate paths are not known by all the routers of the AS [9].

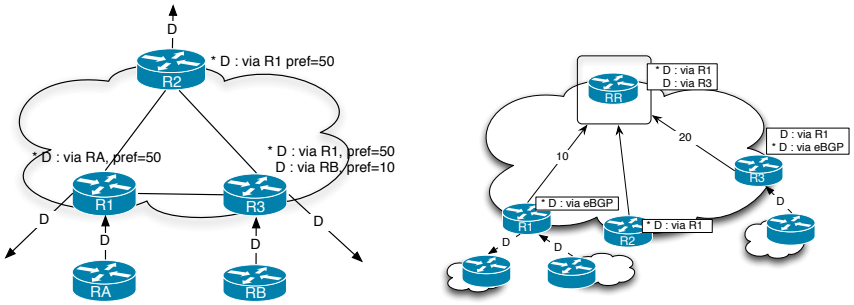
In this paper, we analyse Route Views data to show that an important number of BGP withdraws are probably due to an insufficient alternate paths propagation in iBGP. We propose an incrementally deployable solution that can be used in an AS to ensure that its BGP routers will not propagate a withdraw to neighboring ASes when an alternate path is already known by another BGP router inside this AS. This problem has been identified in [8] as one of the main factors that causes interdomain transient losses of connectivity.

The paper is organised as follows. First, we explain in Sect. 2 how iBGP is used in large ASes. In the next section, we evaluate the number of BGP withdraws that are transienty sent by routers of large ASes although an alternate path is known by another router of this AS. The fourth section details the impact of the internal BGP organization on the propagation of those withdraws. In the fifth section, we present our solution for preventing those unnecessary withdraws. The penultimate section is a review of the related work, and we finish by a conclusion.

2 iBGP Organizations

There are two types of BGP sessions : external BGP (eBGP) sessions between routers belonging to two different Autonomous Systems (AS) and internal BGP (iBGP) sessions established between routers belonging to the same AS. Over an eBGP session, a router announces one path towards each prefix according to its routing policies. Common policies are Customer-Provider and Shared-Cost [10] : Routes learned from customers are advertised to all peers while routes learned from providers and shared-cost peers are only advertised to customers. When an AS contains more than one BGP router, its BGP routers must exchange BGP routes among themselves over iBGP sessions. If the AS is small, these iBGP sessions are usually organised as a full-mesh, i.e. each BGP router has one iBGP session with each other router of the AS. Figure 1(a) shows an AS with a full-mesh of iBGP sessions. Over an iBGP session, each router only advertises the best routes that it learned over eBGP sessions. A BGP router does not advertise over an iBGP session a route that it learned over another iBGP session. The main drawbacks

of using a full-mesh of iBGP sessions are that $\frac{n\times(n-1)}{2}$ iBGP sessions need to be established in an AS with n BGP routers. Moreover, each BGP router receives all the best eBGP paths learned by the AS. This increases the load on all routers inside the AS as they need to process and store a large number of paths.



(a) Full-Mesh (b) Route Reflection

Fig. 1. iBGP organizations

In large ASes containing hundreds or even thousands of BGP routers, it is impossible to use a full-mesh of iBGP sessions. Such large ASes can rely on two possible iBGP organisations : confederations and route reflectors (RR). We focus on route reflection, which is the most widespread iBGP organization, but our solution is also applicable to confederations. A RR is a BGP router that is allowed to advertise over some iBGP sessions the routes that it received over other iBGP sessions. Figure 1(b) show an AS using one route reflector. In most ASes, each edge router has one iBGP session with two different RRs for redundancy. In medium-sized networks, a full-mesh of iBGP sessions is used among the RRs. In larger networks, a hierarchy of RRs is used and only the top-level RRs are interconnected by using a full-mesh of iBGP sessions [9].

We evaluated the cost of using an iBGP full-mesh in GEANT, the pan-european research network, and in a Tier-1 ISP. Geant has 23 routers, each of them having 22 iBGP sessions and about four routes for each prefix in their BGP tables. The Tier-1 ISP uses a two-level hierarchy of Route Reflectors. If it used an iBGP full-mesh instead, the number of iBGP sessions to be maintained by border routers would increase by a factor of 50. With Route Reflection, those routers are RR-clients of two RRs and typically receive two paths for each prefix, one from each of their Route Reflectors. Those two paths are often identical [9], i.e they have the same nexthop address. With a full-mesh, the total number of paths that they have to maintain in their routing tables would be three times higher than with Route Reflection, because in that AS, on average, six different paths are known for each prefix. The number of BGP messages would also be accordingly higher.

When an interdomain link fails, the failure is notified either by the IGP protocol or by the sending of BGP withdraws. The first case occurs when the nexthop address of those BGP routes is the interface to the router from the neighboring AS. It is thus

advertised by the IGP protocol. When the link fails, that nexthop address becomes unreachable in the IGP and the corresponding BGP routes are removed from the BGP tables. If the nexthop address of the eBGP learned routes is the loopback of a local BGP router, the failure is not learned via the intradomain protocol but from BGP withdraws received over iBGP sessions. This is for example the case in Fig. 1(a) : For $R2$, the nexthop of destination D is local router $R1$ and not router RA. Failure detection is faster in the first case, as it relies upon the convergence of the intradomain protocol.

3 Evaluation of the Number of iBGP-caused Withdraws

The propagation of unnecessary BGP withdraws by an AS is responsible for transient losses of connectivity [8, 11]. In this section, we present an analysis of RouteViews BGP feeds that evaluates the number of occurences of such BGP withdraws.

We define a withdraw as iBGP-caused if it was sent by a router of some AS but at least one other router of the same AS did not send a withdraw for the same destination during the same period of time. Indeed, if the second router does not send a withdraw, this means that either it uses another path, or that it knew an alternate path to replace the withdrawn one. In such a situation, at least two paths are available inside the AS, but the alternate path is not known by all routers.

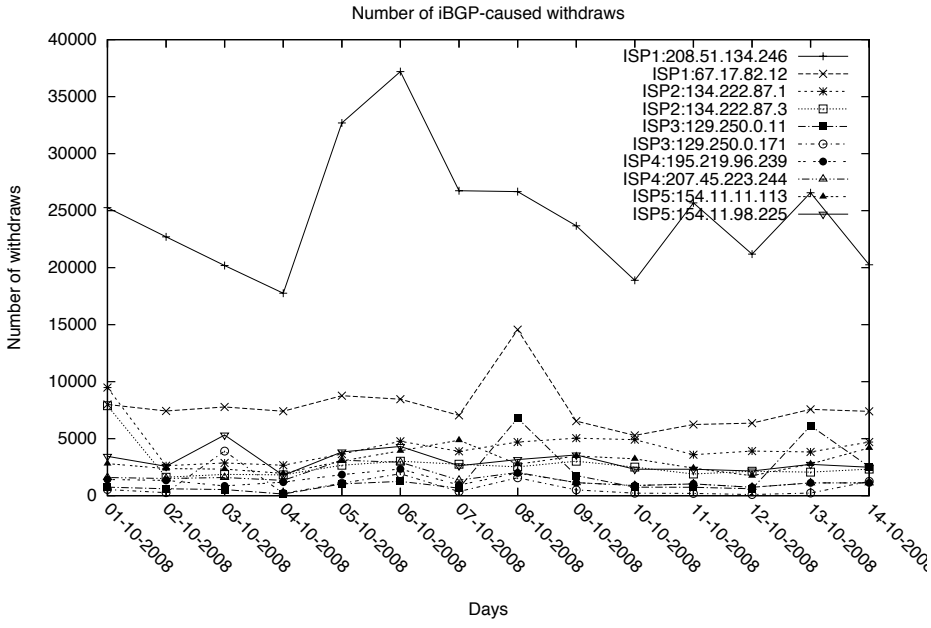


Fig. 2. Number of iBGP-caused BGP withdraws

For our evaluation, we took the BGP data from the first two weeks of October 2008 on the RouteViews Oregon collector, and considered the BGP messages received from

pairs of routers belonging to the same AS. First, we filter all BGP messages received during reboot periods using the BGPMCT algorithm [12]. Second, we classify a BGP withdraw for a destination as iBGP-caused if it is seen on one session with an AS while the other router of this AS has a stable route, i.e. no withdraw for that destination is seen on the session with the other router during 30 seconds before and after the withdraw. This is an upper bound to the propagation time of the withdraw for the path inside an AS, if we assume that, at worst, the withdraw has to cross a whole two-levels iBGP hierarchy between two edge routers, which gives 5 BGP hops.

Figure 2 shows that most of the routers send several thousands of iBGP-caused withdraws per day. On a per-hour basis, results show peaks of more than 2000 iBGP-caused withdraws per hour. Variations between the results for different routers are probably due to different iBGP configurations, but we don't have information about the organizations of the observed ASes. Still, this analysis shows that for all the routers that we analysed, the number of iBGP-caused withdraws is important, and reducing this particular churn would help reducing transient losses of connectivity in the Internet.

4 iBGP Organization and Withdraws Propagation

In this section, we explore the BGP withdraw propagation, and identify that path diversity is the key for blocking this propagation. First, we analyse the problem at the AS level, then we focus on the influence of the iBGP organization on withdraw propagation.

To prevent the unnecessary announcement of a local failure to the entire Internet, the corresponding BGP withdraw must be stopped as close as possible to the failure. We call Withdraw-Blocking a router or an AS that is able to stop the propagation of a withdraw message.

Definition 1. *An AS is said to be* ***Withdraw-Blocking*** *for a destination D if that AS advertises D on at least one eBGP session and does not propagate a BGP withdraw to a neighbor not advertising D itself, upon reception of a BGP withdraw for its primary path towards that destination.*

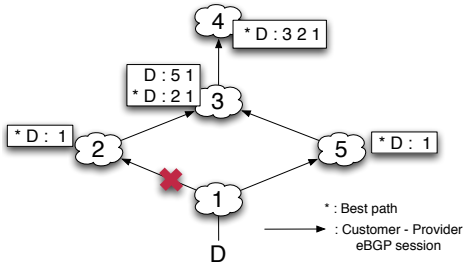


Fig. 3. Withdraw blocking AS

On the topology of Fig. 3, $AS3$ is Withdraw-Blocking for destination D. For example, if the link between $AS1$ and $AS2$ fails, the withdraw is propagated by $AS2$ to

$AS3$. $AS3$ knows the alternate path via $AS5$, such that it can advertise this alternate path to $AS4$ instead of a propagating the withdraw. A withdraw is still sent to $AS5$, but as this neighbor uses and advertises the alternate path, this withdraw won't result in any connectivity loss.

An AS must know an alternate path to reach the destination in order to be withdraw-blocking. However, this is not sufficient, as the AS must forward this alternate path to its neighbor to replace the withdraw message. Policies can prohibit the announcement of the alternate path on some eBGP sessions [10]. Therefore, we define a new property for an alternate path :

Definition 2. *Let S_{P_X} be the set of eBGP sessions on which a path P_X would be advertised if it were the only path available in the AS. A path P_A to destination D is* ***export-policy compliant*** *(EPC) with another path P_B to the same destination if S_{P_B} is included in S_{P_A}.*

In the following subsections, we give conditions for an AS to be withdraw-blocking, first at the AS level (i.e. with ASes composed of a single router). Then, we describe the conditions in the context of ASes composed of multiple routers.

4.1 Withdraw Propagation Prevention with Blackboxed ASes

When modeling an AS as a single router, that "router" knows about all the paths to a given destination learned by the AS. In this case, the conditions to be Withdraw-Blocking are the following :

Theorem 1. *An AS is withdraw-blocking for a destination if and only if it knows an export policy compliant alternate path for its primary path to that destination.*

When the policies used in the AS are the classical routing policies [10], this theorem can easily be proven :

Proof. Two cases must be considered. First, if a withdraw is received from a provider or a shared-cost peer, the only sessions on which the AS was advertising the destination D are customer sessions. In this case, any other path is export-policy compliant, and the AS advertises the alternate path to its customers The second case is when the withdraw is received from a customer. If there exists an alternate path via a peer or a provider, the alternate path is advertised to the customers, but as it is not export-policy compliant, it cannot be advertised over session with peers or providers. The destination is thus withdrawn on those sessions. If the alternate path is learned from the same or another customer, this path is export-policy compliant. As customer path should be preferred over peers and providers paths [10], it will be selected as best once the primary path is withdrawn. The propagation of this customer alternate path is not constrained by policies and an update is sent instead of a withdraw.

4.2 Withdraw Propagation Prevention at the Router Level

In the previous subsection, we analysed the withdraw-blocking property of ASes containing only one router. Real ASes usually contain multiple routers, connected together

by iBGP sessions. When two paths for a given destination are available inside an AS, they are learned over two different eBGP sessions. However, due to the iBGP organization, there can be several routers in the AS that are not necessarily aware of the alternate path [9]. Thus, even if the AS is Withdraw-Blocking, some of its routers may not be Withdraw-Blocking themselves.

For example, consider an AS with a full-mesh of iBGP sessions. If a destination D is learned over two eBGP sessions on two different routers with the same local preference, MED and AS path length, all routers receive both paths. No withdraw is propagated in case of failure of one of those paths. However, if one path has a higher local preference than the other, the alternate path is hidden at the router that received it, because that router prefers the other path. On Fig. 1(a), the path via RB is not advertised by $R3$. If the best path fails, i.e. the path via RA, $R1$ and $R2$ will send withdraws for D on their eBGP sessions.

If Route Reflection is used, the situation is even more problematic [9]. In this case, even if two paths have the same local preference, one of them can be stuck in the Route Reflector, as shown on Fig. 1(b). All RR clients except $R3$ only know the path via $R1$. At least $R1$ will send a withdraw outside the AS if its primary path fails. If the failure is learned via BGP, the BGP withdraw can be blocked by the Route Reflector, because it knows the alternate path via $R3$ and sends to $R2$ and $R3$ an update containing that path instead of a withdraw. However, if the information that the nexthop is not reachable anymore is propagated inside the AS by the intra-domain protocol faster than the BGP messages, all routers that do not have any alternate path will send a withdraw outside the AS even if the Route Reflector does not send withdraws.

Based on this example, we can extend theorem 1 to ASes containing several routers to give a sufficient condition for the Withdraw-Blocking property :

Theorem 2. *An AS is withdraw-blocking for destination D if all routers of the AS know at least one alternate path to D that is export-policy compliant with their primary path.*

Proof. If all routers of the AS have at least one export-policy compliant alternate path, any router that receives a withdraw is able to send an update with the alternate path instead of propagating the withdraw for the primary path on its eBGP sessions. Withdraw propagation is then blocked directly at the border of the AS. Also, when an external nexthop fails, all routers that learn the failure via the IGP have an alternate path, and none will send a withdraw.

Autonomous systems often connect with each other using multiple links [13]. Inside an AS, export-policy compliant paths are then usually available for destinations advertised by multi-connected neighbors. That makes the AS Withdraw-Blocking for these neighbors. However, as explained earlier, local preference settings or iBGP organization prevent this diversity to be propagated to all routers of the AS. It has been evaluated that, in a Tier-1 AS using a two levels' hierarchy of Route Reflectors, most routers typically know only a single path towards a destination [9]. In such an AS, a withdraw can easily be propagated outside the AS even if diversity is available.

5 Blocking Withdraw Propagation outside an AS

Avoiding unnecessary BGP withdraw propagation should allow ISPs to improve the stability of the prefixes advertised by their customer in case of link failure. Furthermore, providing a solution to this problem is affordable, as it can be tackled within the iBGP organization.

When a full-mesh of iBGP sessions is used, it is easy to provide diversity to all routers. Diversity is stuck in a router when there is a better iBGP path (i.e. higher local preference, lower MED or shorter AS path) in the AS. If the advertisement rule is modified such that a router announces its best eBGP-learned path for each destination to its iBGP peers, up to one path per router is propagated in the AS. This mechanism is called Best-External [14].

When an AS uses Route Reflection, it is also possible to prevent withdraw propagation. Using Best-External with Route Reflection is not sufficient, although the best external paths are advertised to Route Reflectors. They do not propagate this diversity further in the network because they only advertise one path per prefix. Modifying the BGP protocol for advertising several paths for each prefix is a solution that has been proposed at the IETF [15]. This allows for a perfect diversity propagation thus achieving the Withdraw-Blocking property, but as it implies increased memory usage and number of BGP messages for exchanging those paths, it is not suitable for ASBR with limited resources.

We propose a lighter solution that would allow routers to propagate the information about the existence of alternate paths without modifying BGP itself. Upon reception of a BGP withdraw, a router that knows that an alternate path exists in the AS can wait until iBGP has converged before sending a withdraw over its eBGP sessions. The AS becomes thus Withdraw-Blocking without requiring its routers to store all BGP routes learned by the AS in their memory.

The principle of our solution is that, whenever a router knows an alternate path, it tags a special BGP community `PATH_DIVERSITY` to the primary path when it advertises it to its iBGP peers, including the one from which the path has been learned if that path comes from iBGP. This is needed because the router that sent the primary path also needs to learn the existence of the backup path. The primary path is then propagated in the AS with the `PATH_DIVERSITY` community. Legacy BGP routers that do not support the community simply propagate the path with the community following classical iBGP rules, without taking its signification into account. The `PATH_DIVERSITY` community is removed when the path is advertised over eBGP sessions.

When the primary path is withdrawn, all routers that support the community and do not have an alternate path themselves will not propagate the withdraw on their eBGP session. Instead, they start a timer and re-advertise the route for the withdrawn path with a local-preference of 0 in iBGP. We thus allow the path to stay temporarily in the routing table of the router. This does not prevent traffic losses, as explained later, but blocks withdraw sending. Also, the routers that do not support the `PATH_DIVERSITY` community will not remove the primary path when receiving the advertisement with the low local-preference and won't send BGP withdraws before receiving the alternate path. With this local-preference value, alternate paths will be preferred over the primary by routers that know them and they will be propagated in the AS [16].

If the timer expires and no alternate path has been received, the router sends the BGP withdraw on its eBGP sessions. The timer is needed if the alternate path is withdrawn shortly after the primary, which can happens when both paths are impacted by the same failure. In that case, the alternate path cannot be propagated in the AS even if the `PATH_DIVERSITY` community has been tagged to the primary path. The timer prevents the BGP convergence to be blocked, waiting for an alternate path that doesn't exist anymore. A suitable value for the timer should be established by evaluating the iBGP convergence time. This value will typically depend on the type of iBGP organization of the AS, and the number of primary paths that can be impacted by a given eBGP link failure.

In the example of Fig. 1(b), R3 tags the path learned via $R1$ with the community, and sends it back to the Route Reflector, which in turn advertises it to all its clients including $R1$. Thanks to this community, $R1$ knows that there exists an alternate path, and if the primary path is withdrawn, it will not send a BGP withdraw on its eBGP sessions. Instead, it will wait until the Route Reflector advertises the alternate path via $R3$.

Algorithm 1 Update reception

```
1:  if BGP path received then
2:      Run decision process
3:      {C}heck diversity
4:      if alternate path exists in Adjribins then
5:          Tag diversity community to the best path, depending on the policies applied to the alternate path.
6:      end if
7:      if Best path changed then
8:          Propagate new best path on eBGP sessions and iBGP sessions (including the originator of the path)
9:      end if
10:     if Best path unchanged, but a community has been tagged then
11:         Advertise on iBGP sessions, including originator of the best path.
12:     end if
13: end if
```

However, as such, the mechanism is not sufficient to ensure the Withdraw-Blocking property of the AS. Indeed, the first alternate path received during the convergence is not necessarily export-policy compliant with the primary one. In this case, the router will have to send a BGP withdraw on the eBGP sessions over which that alternate path cannot be advertised. We refine our solution to face this issue by relying on two community values, `EPC_DIVERSITY` and `NON_EPC_DIVERSITY`. The procedure for tagging those diversity communities is explained in algorithm 1 : A router tags a path with either community depending on whether its alternate path is export-policy compliant with the primary or not. The export-policy compliance can be easily computed by a router if the paths are tagged with a community that identifies their origin [17], i.e. if they come from a customer or from a peer or provider. This is a good practice rule that is often used. The router then readvertises the path to its iBGP neighbors, including to the one from which it was learned.

Algorithm 2 is applied when a router receives a BGP withdraw. The principle is that when a router receives a BGP withdraw for a path tagged with one of the communities,

Algorithm 2 Withdraw reception

```
1: if BGP withdraw received from eBGP session or BGP nexthop becomes unreachable then
2:     Run decision process
3:     if best path unchanged then
4:         check diversity, update communities and readvertise over iBGP if needed
5:     else
6:         if best path is tagged with EPC_DIVERSITY then
7:             if Export-policy compliant path available in Adjribins then
8:                 Propagate alternate path as new best path over iBGP sessions and eBGP sessions
9:             else
10:                Set local-preference of the path to 0
11:                Wait until export-policy compliant path is received, or timer expires
12:                if Timer expires then
13:                    Propagate withdraw
14:                else
15:                    Propagate alternate path as new best path over iBGP sessions and eBGP sessions
16:                end if
17:            end if
18:        else if best path is tagged with NON_EPC_DIVERSITY then
19:            if alternate path is available in Adjribins then
20:                Propagate alternate path over iBGP sessions and over policy-compliant eBGP sessions
21:                Propagage withdraw over non policy-compliant eBGP sessions
22:            else
23:                Set local-preference of the path to 0
24:                Wait until any alternate path is received, or timer expires
25:                if Timer expires then
26:                    Propagate withdraw
27:                else
28:                    Propagate alternate path as new best path over iBGP sessions and over policy-compliant eBGP sessions
29:                    Propagage withdraw over non policy-compliant eBGP sessions
30:                end if
31:            end if
32:        else
33:            Act as usual
34:        end if
35:    end if
36: end if
```

and for which it does not have an alternate path, it does not send any BGP withdraw over eBGP sessions. Instead, it waits until it receives that alternate path. If, during the convergence, a first alternate path that is not export-policy compliant is learned while the path is tagged with `EPC_DIVERSITY`, the router still waits for the export-policy compliant path instead of sending withdraws. When common policies are used [10], the export-policy compliant path will finally be selected as best, and no BGP withdraw is sent over eBGP sessions.

On Fig. 4, RR1 knows an export-policy compliant path via RB, so it adds the community `EPC_DIVERSITY` to the BGP route. RR2 also has diversity for that path. As its alternate path is not export-policy compliant (this is a path received from a provider while the primary comes from a customer), $RR2$ also tags the community `NON_EPC_DIVERSITY`. All routers know that diversity is available, and the AS is Withdraw-Blocking. For example, if the link between RA and $R1$ fails, $R3$ has no diversity but knows that an export-policy compliant path is available, so it does not advertise a withdraw to its eBGP peer. When $RR2$ learns the failure via the IGP, it will not yet send an update for D with the path via RC, because it is not export-policy compliant. Instead, it waits until it receives the export-policy compliant path. Eventually,

$RR2$ then $R3$ will learn the alternate path via RB, and $R3$ can send an update with the export-policy compliant path on its eBGP session.

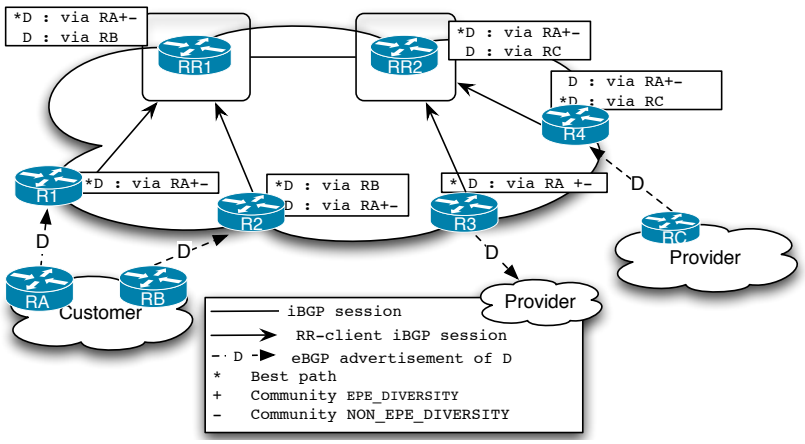


Fig. 4. Announcing diversity in a community

BGP convergence Using those communities slightly increases the number of BGP messages exchanged during the initial convergence, as an additionnal update is emitted when the route is tagged with a diversity community. In the worst case, two additionnal updates will be emitted by a router, one when a non export-policy compliant alternate path is known to exist, and a second when the existence of an export-policy compliant alternate path is learned. Also, upon failure of an alternate path, a few BGP messages are also exchanged to update the communities of the primary path. However, those BGP messages will not be announced outside the AS, hence the small message overhead is limited to the AS.

The diversity communities also do not impact routing stability : Tagging the diversity communities is a deterministic process that does not lead to routing loops. Indeed, when the paths to a destination are stable, once a diversity community has been tagged to a route, it is not removed as long as there is a corresponding alternate path in the AS. Sending the tagged path back to the sender or the original path also does not result in routing loops. As a path is at most tagged twice, it is sent back to the sender at most twice and then the routing state becomes stable. Also, if the alternate path fails, the router that tagged the primary path stops re-advertising it with the backup community, and the tagged path is replaced by the original path in all routers after iBGP convergence.

Impact on the data plane When a router receives a withdraw for a destination and waits for the alternate path, it doesn't know any other nexthop to which send the traffic until it receives the new path. The traffic might then be dropped during the iBGP convergence. However, even if it cannot prevent the local loss of connectivity, waiting for the new path before sending the withdraw on eBGP reduces the losses of connectivity that would occur further in the Internet due to the unnecessary withdraw propagation.

6 Related Work

As explained in the introduction, the churn that affects BGP has several causes. The first one is the path vector nature of BGP. This, combined with routing policies, leads to path exploration as explained by Gao et al. among others in [18]. Several solutions have been proposed to reduce the impact of path exploration in the global Internet. These solutions rely on the utilization of new BGP attributes. The most complete ones are BGP RCN proposed by Pei et al. [19] and EPIC proposed by Chandrashekar et al. [20], but are not currently deployed. Our solution complements these approaches.

The current iBGP organizations are known to be imperfect [21]. Several researchers have proposed solutions to improve them. One approach is to centralise all routing decisions in a Routing Control Platform that can be considered as a super Route Reflector RCP [22]. Another approach is to extend iBGP to allow each router/RR to advertise several paths towards each destination [15]. This iBGP extension allows all BGP routers inside an AS to learn several paths towards each destination and thus block the propagation of withdraws, achieving the same objective as our communities. Furthermore, as the backup paths are propagated and not only the information about their availability, connectivity losses are also prevented. This extension also allows for other utilizations of additionnal paths, such as multipath routing. This is definitely a very promising solution for the future, but currently, as this mechanism increases the number of BGP messages and the memory required to store all the additional paths, all routers cannot support it. In the meantime, our communities can be used to prevent withraw propagation, as they do not require any change to the BGP syntax and could be deployed incrementally .

7 Conclusion

In this paper, we have first explained that the iBGP organisation used in large ASes reduces the number of paths learned by each router. When a link fails or a path is withdrawn, BGP routers inside an AS may send an unnecessary BGP withdraw. This causes transient losses of connectivity. We proposed a solution that allows routers to know if there is an alternate path for each prefix inside the AS. When a link fails or a BGP withdraw is received, BGP routers will block the propagation of withdraws for prefixes for which an alternate path is known in the AS.

Our further work is to evaluate the convergence time of iBGP with and without using our solution.

References

1. Rekhter, Y., Li, T., Hares, S.: A border gateway protocol 4 (BGP-4). Internet RFC4271 (2006)
2. Meyer, D., Zhang, L., Fall, K.: Report from the IAB workshop on routing and addressing. RFC4984 (2007)
3. Bonaventure, O., Filsfils, C., Francois, P.: Achieving sub-50 milliseconds recovery upon BGP peering link failures. IEEE/ACM Trans. Netw. **15** (2007) 1123–1135
4. Wu, J., Mao, Z.M., Rexford, J., Wang, J.: Finding a needle in a haystack: pinpointing significant BGP routing changes in an IP network. In: NSDI'05, Berkeley, CA, USA, USENIX Association (2005) 1–14
5. Wang, F., Gao, L., Wang, J., Qiu, J.: On understanding of transient interdomain routing failures. In: ICNP '05, Washington, DC, USA, IEEE Computer Society (2005) 30–39
6. Oliveira, R., Zhanf, B., Pei, D., Izhak-Ratzin, R., Zhang, L.: Quantifying path exploration in the internet. In: Internet Measurement Conference, Rio de Janeiro, Brazil (2006)
7. Mao, Z.M., Govindan, R., Varghese, G., Katz, R.: Route flap damping exacerbates internet routing convergence. In: ACM SIGCOMM'2002. (2002)
8. 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: SIGCOMM '06, New York, NY, USA, ACM (2006) 375–386
9. Uhlig, S., Tandel, S.: Quantifying the impact of route-reflection on BGP routes diversity inside a tier-1 network. In: IFIP Networking 2006, Coimbra, Portugal (2006)
10. Gao, L., Rexford, J.: Stable internet routing without global coordination. IEEE/ACM Trans. Netw. **9** (2001) 681–692
11. Kushman, N., Kandula, S., Katabi, D., Maggs, B.: R-BGP: Staying Connected in a Connected World. In: NSDI'07, Cambridge, MA (2007)
12. Zhang, B., Kambhampati, V., Lad, M., Massey, D., Zhang, L.: Identifying bgp routing table transfers. In: MineNet '05: Proceedings of the 2005 ACM SIGCOMM workshop on Mining network data, New York, NY, USA, ACM (2005) 213–218
13. Feamster, N., Mao, Z.M., Rexford, J.: BorderGuard: Detecting Cold Potatoes from Peers. In: Internet Measurement Conference, Taormina, Italy (2004)
14. Marques, P., Fernando, R., Chen, E., Mohapatra, P.: Advertisement of the best-external route to iBGP. Internet Draft, draft-marques-idr-best-external-00, work in progess (2008)
15. Walton, D., Retana, A., Chen, E.: Advertisement of Multiple Paths in BGP. Internet draft, draft-walton-bgp-add-paths-05.txt, work in progress (2006)
16. Francois, P., Decraene, B., Pelsser, C.: Graceful eBGP Session Shutdown. Internet draft, draft-francois-bgp-gshut-00.txt, work in progress (2008)
17. Meyer, D.: BGP Communities for Data Collection. RFC 4384 (Best Current Practice) (2006)
18. Gao, L., Wang, F.: The extent of as path inflation by routing policies. In: Global Internet 2002. (2002)
19. Pei, D., Azuma, M., Nguyen, N., Chen, J., Massey, D., Zhang, L.: BGP-RCN: Improving BGP convergence through Root Cause Notification. Computer Networks **48** (2005) 175–194 2005.
20. Chandrashekar, J., Duan, Z., Zhang, Z., Krasky., J.: Limiting path exploration in BGP. In: IEEE INFOCOM, Miami, Florida (2005)
21. Griffin, T., Wilfong, G.: On the correctnes of iBGP configuration. In: SIGCOMM'02, Pittsburgh, PA, USA (2002) 17–29
22. Caesar, M., Caldwell, D., Feamster, N., Rexford, J., Shaikh, A., van der Merwe, J.: Design and implementation of a routing control platform. In: NSDI'05, Berkeley, CA, USA, USENIX Association (2005) 15–28