# Keep your Communities Clean: Exploring the Routing Message Impact of BGP Communities

Thomas Krenc
Naval Postgraduate School
tkrenc@nps.edu

Robert Beverly
Naval Postgraduate School
rbeverly@nps.edu

Georgios Smaragdakis
TU Berlin
georgios.smaragdakis@tu-berlin.de

## ABSTRACT

BGP communities are widely used to tag prefix aggregates for policy, traffic engineering, and inter-AS signaling. Because individual ASes define their own community semantics, many ASes blindly propagate communities they do not recognize. Prior research has shown the potential security vulnerabilities when communities are not filtered. This work sheds light on a second unintended side-effect of communities and permissive propagation: an increase in *unnecessary* BGP routing messages. Due to its transitive property, a change in the community attribute induces update messages throughout established routes, just updating communities. We ground our work by characterizing the handling of updates with communities, including when filtered, on multiple real-world BGP implementations in controlled laboratory experiments. We then examine 10 years of BGP messages observed in the wild at two route collector systems. In 2020, approximately 25% of all announcements modify the community attribute, but retain the AS path of the most recent announcement; an additional 25% update neither community nor AS path. Using predictable beacon prefixes, we demonstrate that communities lead to an increase in update messages both at the tagging AS and at neighboring ASes that neither add nor filter communities. This effect is prominent for geolocation communities during path exploration: on a single day, 63% of all unique community attributes are revealed exclusively due to global withdrawals.

## CCS CONCEPTS

• **Networks** → **Network measurement**; *Network protocol design.*

## KEYWORDS

BGP, Communities

**ACM Reference Format:**
Thomas Krenc, Robert Beverly, and Georgios Smaragdakis. 2020. Keep your Communities Clean: Exploring the Routing Message Impact of BGP Communities. In *The 16th International Conference on emerging Networking EXperiments and Technologies (CoNEXT '20), December 1–4, 2020, Barcelona, Spain.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3386367.3432731



*CoNEXT '20, December 1–4, 2020, Barcelona, Spain*

ACM ISBN 978-1-4503-7948-9/20/12…$15.00
https://doi.org/10.1145/3386367.3432731

## 1 INTRODUCTION

Border Gateway Protocol (BGP), the Internet's inter-domain routing protocol, is fundamental to the operation, policies, and economics of the network. Unsurprisingly, the real-world behavior of BGP has been subject to intense scrutiny [20, 31, 37]. As an extensible protocol, BGP and its usage have evolved in response to operator needs. BGP communities [10, 26] are one such example of an optional attribute that adds new meta-information to routing messages.

Communities are used to conveniently tag (an aggregate of) prefixes to enable particular actions or policies. However, community values and semantics are not standardized – the meaning of a specific community value is defined by individual autonomous systems (ASes). Benoit *et al.* first defined a taxonomy where, broadly speaking, communities are used to tag received prefixes to aid an AS's internal routing decisions, or added to outbound announcements as a convenient signaling mechanism [10]. Today, communities are known to encode location identifiers [15, 17] express policy preferences upstream [18], enable selective advertisement in Internet Exchange Points (IXPs) [19, 34], and as a DDoS mitigation signal [9, 16] (e.g. BGP blackholing [6, 23]).

BGP communities have been widely adopted. Streibelt *et al.* found a 250% increase in the number of unique communities and a 200% increase in the number of ASes that use communities as seen in BGP advertisements between 2010 and 2018 [41]. Giotsas *et al.* [14] report that around 50% of IPv4 and 30% of IPv6 BGP announcements in 2016 include at least one location-tagged BGP community.

Despite the rich literature on BGP and BGP communities, prior work has not investigated the unintended impact of communities on the *volume* of BGP message traffic in the wild. Isolating and decoupling the traffic impact of communities from other mechanisms and variables in the complex Internet without ground-truth configurations is challenging. In this work, we take a first step towards this larger goal by examining changes in AS paths and communities in billions of update messages at 500+ peers over 10 years. We find that updates with no path change are common throughout the entire measurement period and are rooted in widespread community deployment, increasingly interconnected networks, and lack of community filtering. More specifically, we find:

(1) Around 50% of announcements in March 2020 signal no path change, while half of these exhibit a community change. We consider these announcements *unnecessary*.
(2) In laboratory experiments and in the wild, we show that community geo-tagging in combination with missing or ineffective filtering can lead to an increase of update messages.
(3) Among 10 tested router implementations, all but two forward communities by default. Also, 7 routers generate duplicates due to filtering communities at egress.

(4) By utilizing beacon prefixes we show that more than 60% of all encoded information in community attributes is revealed during global withdrawals, as a result of path exploration.

We publicize and validate our work through mailing lists and online documentation [7]. Our findings resonated with the community, in particular with router developers, and helped identify a source of duplicates in a current popular BGP implementation. For reproducibility we publish code to identify unnecessary announcements and router configuration of all tested routers. Our findings afford a better understanding of routing instabilities in the Internet, unnecessary load on the system, and may help foster detection of anomalous communities in the future. We conclude by discussing implications on routing message archival and suggest future work.

## 2 BACKGROUND AND RELATED WORK

This section provides details of BGP relevant to understanding our work on communities, as well as a summary of prior research. We assume working familiarity with BGP; see [37] for a broad overview.

**Path Exploration:** The BGP decision process is complicated, involving iBGP, eBGP and IGP interaction, is governed by individual network policies, and beholden to BGP implementation particulars. There is a basic tension between propagating reachability information quickly and sending it prematurely, i.e., before the AS has converged on a new best state. In practice, implementations and configurations of BGP often delay sending messages. Thus, updates often occur in bursts.

Several prior works analyze network stability, path exploration, and the trade off in withholding updates [31]. Mechanisms such as route dampening and MRAI timers [8] have been explored, but may offer sub-optimal performance in reacting to routing events [43]. Thus, these mechanisms are selectively deployed. Indeed, we show that path exploration, combined with BGP community use, is a significant contributor to BGP update traffic.

**Communities:** BGP messages are relatively simple and include prefix *updates* (often termed an "announcement") as well as prefix *withdrawals* (indicating that the prefix should be removed from the routing table). In addition, BGP messages can include multiple optional attributes, among them the next-hop, MED, and community. Among these, BGP communities are notable because they are transitive – meaning that they are an optional attribute that may be propagated. Communities are the focus of our study. As we will show, not only are BGP communities in common use, but they are a primary contributor to overall BGP message traffic.

BGP communities are simply a 32-bit value, however a common convention is that the upper two bytes encode the ASN of the AS that "owns" the community, while the lower two bytes define the meaning of the community. Because communities have no well-defined semantics, it is up to each individual AS to define the meaning of the lower two bytes corresponding to their community space. Note that while large and extended communities have since been added [22, 40], for instance to accommodate 32-bit ASNs and to communicate additional bits of information, these are in infrequent use. Hence, our study focuses on traditional BGP communities.

Benoit *et al.* provides a taxonomy of BGP communities in [10]. As a contemporary convention, BGP communities can be broadly divided into informational communities and action communities [40]. Informational communities are typically added to ingress routing announcements to tag aggregates of routes in a common way in order for an AS to make internal policy and routing decisions. For instance, a common informational community used by large ASes is to encode the physical geographic location where a prefix is received, e.g. "North America, Dallas, TX."

In contrast, action communities are frequently added to egress announcements to implement in-band signaling to a different AS. For instance, a common use of action communities is the blackhole community which indicates that a provider should stanch traffic for a particular IP or prefix that is experiencing a DDoS attack.

With the growth in BGP community adoption, researchers have in recent years explored community prevalence and security [10, 41], as well as the information they leak about connectivity, attacks, and outages [14, 16]. However, less attention has been given to the unintended impacts of communities on the volume of BGP message traffic in general, and updates in particular – these are the focus of the present research.

**Duplicate Updates:** As with the protocol itself, BGP update behavior has been extensively studied, for instance to understand routing dynamics [25], understand convergence and forwarding behavior [28], quantify path performance [43], and locate origins of instabilities [11]. Of most relevance to our present study are so-called "duplicate" BGP updates, superfluous messages that do not update routing state. First identified by Labovitz in 1998 [24] and believed to be attributable to buggy implementations, Park *et al.* subsequently demonstrated that iBGP/eBGP interaction was the primary cause of these duplicates [30]. As we also find, when a router receives an internal update with a changed attribute, that attribute may be removed or replaced prior to announcing to an eBGP peer, resulting in a duplicate. Hauweele *et al.* later verified in both real and lab experiments that MED, next-hop, and community attribute changes induce these duplicates [21].

While duplicate updates have been a recognized issue for decades, we first show via controlled lab experiments in §3 the propagation, update, and duplication implications of communities specifically. Second, our work attempts to quantify the impact of communities on BGP message traffic in the wild and over time. Our findings in §6 highlight the effects of increased BGP community use on update generation. We show that BGP geolocation communities are a primary source of unnecessary updates, and induce inter-AS message traffic even when communities are filtered.

## 3 CONTROLLED EXPERIMENTS

To validate our findings and inferences, as well as gain a deeper understanding of BGP update root causes, we conduct a series of experiments in a controlled laboratory setting.

For each experiment run, we configure all routers depicted in Figure 1 to use one of the following routing software: Cisco IOS (12.4(20)T), IOS XR (v6.0.1), Juniper Junos (Olive 12.1R1.9), Nokia SR OS (20.7.R2), BIRD (v1.6.6 and v2.0.7) [2], FRRouting (v6.0.2) [13], OpenBGPD (v5.2 and v6.6) [29] or Quagga (v1.2.4) [32]. While Cisco and Juniper routers dominate the core router market, for example BIRD is used in many large IXPs [34, 39]. FRRouting and Quagga are used in RouteViews and RIPE collectors to listen for BGP updates

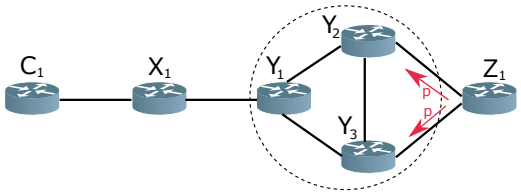


**Figure 1: Laboratory topology to understand conditions generating BGP update messages**

and dump them to Multi-Threaded Routing Toolkit (MRT) [4] files. Our list of routing software is not comprehensive.

We run IOS, Junos and SR OS in emulation. By using these router images, as well as routing daemons, we can gain insight into real-world BGP implementation behavior. Because we find identical behaviors for most of the experiments, we report only on the common behavior and where it deviates.

The lab topology is crafted to test several scenarios, with and without communities and filtering, to understand the conditions that generate BGP update messages and when those update messages are propagated. The topology consists of four ASes: $X$, $Y$, $Z$ and $C$. Router $C_1$ mimics a route collector, while $Z_1$ originates the prefix $p$. The links in the topology correspond to both physical connections and eBGP and iBGP sessions. AS $Y$ has three routers within its network; both $Y_2$ and $Y_3$ peer with AS $Z$.

Prior to running our experiments, we verify that only BGP keep alive messages, i.e., pairwise heartbeats to test liveness, are sent once the network has converged. In the following experiments, we are interested in BGP messages that carry announcements and withdrawals. We use tools like tcpdump and generated logs to inspect this message exchange.

- **Exp1**: We begin without any BGP communities to characterize default behavior. Note that border router $Y_1$ has two paths to reach $p$. In the absence of any policy, the BGP tie breaker selects $Y_2$ as the next hop. Therefore, to induce BGP updates, we disable the $Y_1$ to $Y_2$ link, and perform a packet capture of all messages arriving at the collector $C_1$ and between $X_1$ and $Y_1$.
  Without BGP communities, when $Y_1$ chooses a new next hop of $Y_3$, it sends an update message to $X_1$ even though the AS path has not changed. (Note: Junos and SR OS do not generate duplicates). However, this update is ignored by $X_1$ and not propagated further – no update message is observed at the collector.
- **Exp2**: Next, we consider the common scenario where AS $Y$ implements communities that geographically tag incoming advertisements. $Y_2$ adds community `Y:300` on ingress while $Y_3$ adds `Y:400`. Because $Y_2$ is preferred, and no community filtering is implemented in this network, the collector sees $p$ with `Y:300`. We again disable the $Y_1$ to $Y_2$ link.
  Again, this induces an update message from $Y_1$ to $X_1$. While the AS path is unchanged, this update includes a changed community value of `Y:400`. Because the community value changed (implicit withdrawal), $X_1$ also sends an update which is seen at the collector. Note that while updates sent by $Y_1$ can be due to an internal next-hop change (as in Exp1), in the case of $X_1$ the next-hop does not change. Thus, a change in the community attribute is the sole trigger for the update. (Note: IOS needs explicit configuration to forward communities on eBGP sessions).

**Table 1: Overview of experiments and results.**

| Routing software | Exp1: $Y_1$ sends dups (next-hop change) | Exp2: $X_1$ forwards communities set by $Y_1$ | Exp3: $X_1$ cleans at egress, sends dups | Exp4: $X_1$ cleans at ingress, no dups generated |
|---|---|---|---|---|
| Cisco IOS 12.4(20)T | true | **false** | true | true |
| XR v6.0.1 | true | **false** | true | true |
| Juniper Junos Olive 12.1R1.9 | **false** | true | **false** | true |
| Nokia SR OS 20.7.R2 | **false** | true | **false** | true |
| BIRD v1.6.6 | true | true | true | true |
| v2.0.7 | true | true | true | true |
| FRRouting v6.0.2 | true | true | true | true |
| OpenBGPD v5.2 | true | true | true | true |
| v6.6 | true | true | **false** | true |
| Quagga v1.2.4 | true | true | true | true |

- **Exp3**: We implement community filtering on $X_1$ by configuring it to remove all communities on egress. We again flap the $Y_1$ to $Y_2$ link to generate the update message. Surprisingly, even though $X_1$ is removing communities, it still sends an update to the collector (Note: Junos, SR OS and OpenBGPD v6.6 do not generate duplicates). Note that this update has an unchanged AS path and includes no communities – i.e., it is an arguably unnecessary message.
- **Exp4**: We then repeat experiment 3, but modify $X_1$ to filter communities on ingress from $Y_1$. In this case, the spurious update message is not sent as the communities are not contained in the router's RIB. This shows that we can differentiate between ingress and egress community filtering.

**Summary:** Table 1 summarizes the experiments and results. Among the tested software, by default, only Junos and SR OS prevent duplicates from being generated by, e.g. internal changes or community filtering on egress. OpenBGPD v6.6 suppresses duplicates when the community changes but not when the next-hop changes. Furthermore, all routers generate updates that are triggered only due a change in the community attribute, if communities are not filtered at ingress. Our findings imply that this behavior is transitive. For reproducibility, we publish the relevant configurations for each tested router software [7].

We note that sending updates with no changes contradicts BGP specifications. According to RFC4271 §9.2 [33]: "A BGP speaker SHOULD NOT advertise a given feasible BGP route from its Adj-RIB-Out if it would produce an UPDATE message containing the same BGP route as was previously advertised." In reality, maintaining the Adj-RIB-Out requires keeping significant state which is a major concern for operators when making tradeoffs between convergence speed and resource usage. Also, vendors and developers design their software with different default configurations. While for example Junos maintains the Adj-RIB-Out by default, BIRD requires explicit configuration by the user. The open-source implementations are not only feature rich, but also they can operate on many different hardware architectures with different memory capacities. Thus, default configurations are kept at a minimum setting leaving the responsibility to the user.

**Table 2: Overview $d_{mar20}$ data set**

| | | | |
|---|---|---|---|
| IPv4 prefixes | 1,071,150 | Announcements | 1,008M |
| IPv6 prefixes | 99,141 | w/ communities | 737.0M |
| ASes | 68,911 | uniq. 16 bits | 5,778 |
| Sessions | 1,504 | uniq. AS paths | 43.9M |
| Peers | 581 | Withdrawals | 38.5M |

## 4 DATA SETS

To study the impact of BGP communities on update message propagation in the wild, we use publicly available archived routing traffic from the RouteViews [38] and RIPE RIS [36] BGP collector projects. We obtain all MRT formatted update (251,493) and RIB (9,539) files from all collectors, inclusive of both IPv4 and IPv6 prefixes as well as withdrawals, for a full day every 3 months (2019-03-15, 2019-06-15, 2019-09-15, etc.) across a ten-year span (2010 to 2020). While our analyses are based on update messages only, we use the RIB snapshots to detect peer ASes that do not occur in the update files.

Prior to analyzing the raw data, we first perform basic filtering, cleaning, and normalization, so as to not impart unintentional bias. Using current and historical allocation information from the regional registries, we remove messages that contain an unallocated ASN or prefix at the time of the message. We do not aggregate overlapping prefixes, and we keep all prefixes, regardless of their length. We note that 7 peer ASes do not prepend their ASN to the AS path when advertising routes to the collector, i.e., the *FROM* field in the update message differs from the left-most AS in the AS path. Those ASes are IXP route servers. To avoid overcounting peer ASes and avoid ambiguity when processing the data, we add the ASN of the route server to the AS path. Finally, only 4 BGP collectors (RouteViews: `route-views3, eqix, linx, sfmix`) record update messages at a microsecond granularity, as of March 2015. All other collectors use a single second granularity for received updates. When multiple messages arrive within the same second for these collectors, we preserve the message ordering and assume that each subsequent message arrives 1$\mu$s after the previous.

In the remainder of this paper, we refer to the resulting data set as $d_{hist}$. We use $d_{mar20}$ to point to the most recent data in our measurements, which is March 15, 2020. Table 2 provides an overview of $d_{mar20}$. The data set includes 1,504 sessions across 581 unique peer ASes. The number of BGP sessions at these two collector projects has roughly doubled over the past ten years. We note that not all the peers send updates on any day. In $d_{mar20}$, we find updates for 451 of the 581 peer ASes.

**Routing Beacons:** Routing beacons are prefixes announced and withdrawn at periodic intervals [28]. Beacons can help network operators and researchers investigate the routing system and routing anomalies by providing a source of predictable and known behavior. RIPE operates routing beacons [35] with an update pattern of a single announcement every 4 hours, starting at 00:00 UTC, and a single withdrawal every 4 hours, starting at 02:00 UTC. One specific IPv4 and IPv6 beacon prefix is announced per RIPE route collector.

From 39 available RIPE beacon prefixes, we remove 4 IPv4 beacons that are either not active or too noisy. Then, we select all announcements and withdrawals that are associated with the remaining 35 beacons. We observe 660,567 announcements and 115,892 withdrawals spread over 998 sessions, 354 peers, and 34 collectors. We refer to this subset as $d_{beacon}$.

**Table 3: Announcement types (share in $d_{mar20}$ and $d_{beacon}$)**

| type | observed changes | $d_{mar20}$ | $d_{beacon}$ |
|---|---|---|---|
| *pc* | path + community | 33.7% | 44.6% |
| *pn* | path only | 15.1% | 29.9% |
| **nc** | **community only** | **24.5%** | **13.8%** |
| **nn** | **no change** | **25.7%** | **11.2%** |
| *xc* | path prepending + comm. | 0.3% | 0.2% |
| *xn* | path prepending only | 0.7% | 0.3% |

## 5 ANNOUNCEMENT TYPES

To better understand announcements in our studied data sets, we first group them by the prefix and the BGP session (the `<peer AS, next-hop>` tuple), in arriving order. Then, from one announcement to the next, we look for changes (or no changes) in the community attribute and in the AS path. In the AS path, we further distinguish between a change in the set of ASNs, and a change in path prepending, i.e., inflation or deflation of an identical set; it cannot be both. From three possible observations in the AS path attribute and two in the community attribute, we define six different combinations of two letters to label the *announcement type*: The first letter indicates the AS path (*p* = path change, *n* = no path change, *x* = path prepending), and the second letter indicates the community attribute (*c* = community change, *n* = no community change): *pc*, *pn*, *nc*, *nn*, *xc*, *xn*.

An announcement with a path change only is in the category *pn*. If there is a change in path prepending (the set of ASes are equal), it is in *xn*. If, in addition to the path also the community attribute changes, the announcement is in *pc* (or *xc* in case of path prepending). While we intuitively expect *pn*, *pc*, *xn* and *xc* updates, we also see updates without a path change: *nc* and *nn* cover all announcements with no path change, while the former also includes changes in the community attribute. We note that *nn* also includes two empty community attributes in succession. Also, we acknowledge that the MED attribute for a the `<peer AS, next-hop>` tuple can change towards the collector as a reason for an nn announcement.

**Statistics:** Table 3 provides a break-down of the possible observations in $d_{mar20}$. We note that *nc* and *nn* – the only types that do not include a path change – make up more than half of all announcements (24.5% and 25.7% respectively). The largest group of announcements is of type *pc* with a share of 33.7%, while *pn* announcements constitute 15.1%. The number of announcements that either inflate or deflate a given AS path is negligibly small, contributing around 1%. For comparison, we also provide the share of types in $d_{beacon}$. Here, we see a different distribution. While *nc* and *nn* together contribute 25% to all announcements, *pc* is the most dominant type with a share of 44.6%, followed by *pn* with 29.9% share. Again, *xn* and *xc* are low in numbers. Recall that beacon prefixes provide us with a more controlled view on the update behavior since they are announced and withdrawn at stated intervals. In the wild, however, unpredictable changes at the origin AS can add to the dynamics of update propagation at all downstream paths.

Next, we investigate the longitudinal behavior of announcement types, using $d_{hist}$. In Figure 2, we show the share of the individual constituents on the left y-axis, and the total number of announcements on the right y-axis, over time. First, we note that the overall

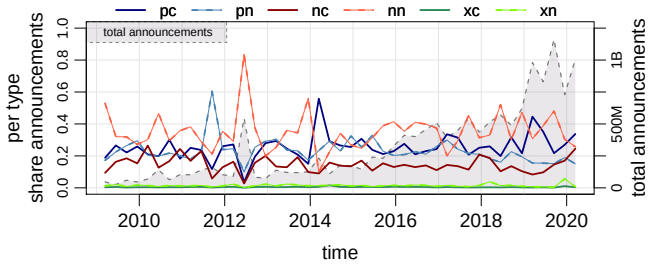


**Figure 2: Share of daily announcements per type and total announcements in $d_{hist}$.**

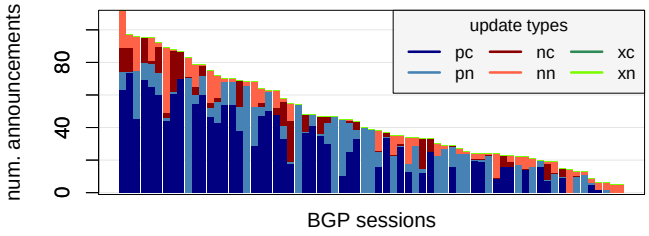


**Figure 3: Announcement types per BGP session for beacon prefix** 84.205.64.0/24**, collector: rrc00, March 15, 2020**

number of announcements show a 40-fold increase from around 26M in 2009 to 1.1B in 2019. Except for the spike[1] in 2012, the number is below 250M until 2015, after which it roughly doubles to 500M in 2018, followed by another doubling in 2019 to 1B. This increase can be explained by the parallel increase of peer ASes that send update messages to the collectors, but also by the overall increase of routing activity. Second, looking at the share of the individual announcement types, we observe some amount of variability in the distribution over time. The standard deviation over all time points ranges between 0.047 and 0.119 for the individual types. The median of (*pc*, *pn*, *nc*, *nn*) announcements over the last ten years is (26%, 24%, 15%, 31%), respectively. We note that $d_{hist}$ is limited to a full day of update data per quarter year. In a separate analysis, using ten consecutive days in February 2020, we find less variability ($\sigma < 0.04$) on a daily basis.

## 6 UNNECESSARY UPDATES

Next, we study communities in update messages with no path change and their impact on update propagation. We focus on announcements and withdrawals for individual beacon prefixes ($d_{beacon}$) visible in BGP sessions over 24 hours.

**BGP Sessions.** We begin by investigating how peers of a routing collector perceive the different announcement types, i.e. *pc*, *pn*, *nc*, *nn*, *xc*, and *xn*. We note that a peer AS sends only the best path via BGP sessions to the collectors. The stacked bar plot in Figure 3 includes each of the BGP sessions of RIPE collector `rrc00`. The sessions are sorted by number of announcements visible for prefix 84.205.64.0/24 $\in d_{beacon}$ and colored to indicate the announcement type. We observe that each session shows a different number

[1]The spike of *nn* activity in June 2012 is an artifact of peer AS821 at the `route-views2` collector sending more than 1220 duplicate announcements for more than 210K prefixes throughout a single day.

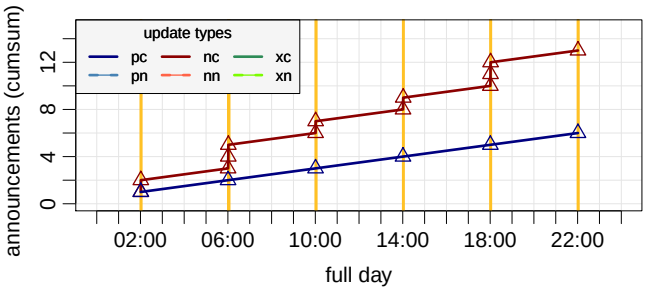


**Figure 4: Announcement types over time with prefix** 84.205.64.0/24 **and AS path (`20205 3356 174 12654`). Geotagging induces *nc* announcements.**

of announcements. But more interesting is the fact that each session shows a diverse distribution of announcements types, despite looking *only* at a single beacon prefix. We characterize the peer ASes later in this section. To this end, the root causes, i.e., why *nc* and *nn* announcements are sent in the first place, are unclear. In the following we take a closer look at those announcement types.

**Community Exploration.** To highlight the conditions that can lead to *nc* announcements in the wild, we present an example using the view of a single BGP session. In Figure 4 we show the cumulative sum of announcements over 24 hours of March 15, 2020. We plot all announcements for the same prefix 84.205.64.0/24 $\in d_{beacon}$ via a single AS path (`20205 3356 174 12654`). Vertical yellow lines indicate the arrival of a withdrawal message for that prefix, confirming the withdrawal interval for routing beacons.

All announcements for this particular route and day show up only during the withdrawal phases, i.e. at around 02:00, 06:00, 10:00, etc. We deduce that this particular route was never a best path during that day (all time best path: `20205 6939 50304 12654`). During the six withdrawal phases we observe a total of 19 announcements: Starting with a *pc* update (6 total), i.e., an announcement with changed path and community, followed by multiple (13 total) *nc*'s, announcements with changing community only. Peer AS20205 does not set any communities. However, it does not clean communities from its neighbor either: The changing communities in *nc* announcements represent encoded ingress locations set presumably by AS3356. We observe a total of 9 locations encoded in 19 announcements: 9 city communities, two country and two geographical regions, i.e., Europe and North America. Per withdrawal phase, the location communities are mostly unique.

Due to distinct location communities attached to a single route, multiple *nc* announcements occur (comparable to Exp2 in §3). Analogously to path exploration, we refer to this behavior as *community exploration*: Instead of multiple paths being announced, multiple communities for a single path are announced. Also, the example above demonstrates that setting communities by one AS, can impact the update behavior of a different AS, if no proper filtering is in place (comparable to Exp4).

**Duplicate Announcements.** Next, we explore a possible reason for the occurrence of *nn* updates. Therefore, we choose a route similar to the previous community exploration example. However, we replace the peer AS with one that removes all communities (in >99% of the cases). Figure 5 shows the cumulative sum of announce-

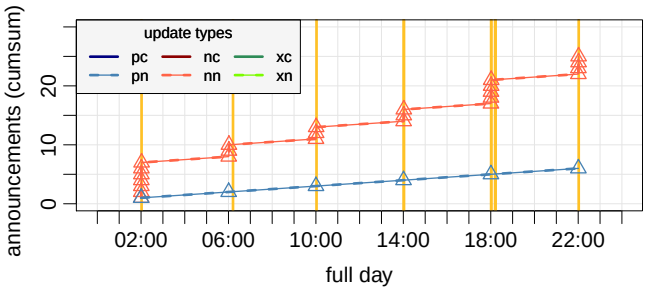


**Figure 5: Announcement types over time with prefix 84.205.64.0/24 and AS path (`20811 3356 174 12654`). Cleaning at egress generates *nn* announcements.**

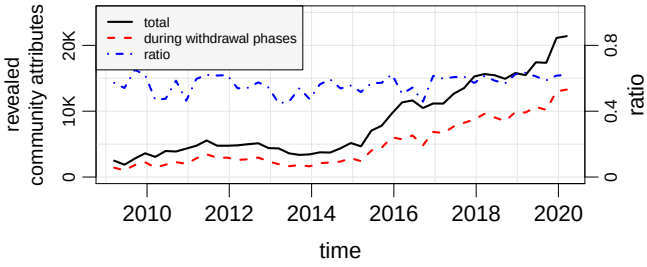


**Figure 6: Revealed unique community attributes during withdrawal phases of all RIPE beacon prefixes over time.**

ments over the day of March 15, 2020. We plot announcements for the same prefix 84.205.64.0/24, but via a different AS path (`20811 3356 174 12654`). Vertical yellow lines represent withdrawal messages for that prefix, again in accordance with the predefined intervals. Again, all 31 announcements occur during the withdrawal phases. Also, the phases begin with a path change (6 total), here *pn*, followed by a series of *nn* announcements (25 total).

Deduced from our previous observations, we speculate that during the withdrawal phase AS20811 simply reannounces multiple *nc*'s from AS3356 (as an implicit withdrawal) and removes the existing communities prior to announcing, thus inducing *nn* announcements. Note, we have demonstrated such behavior in lab experiments (Exp3). We manually re-visit the raw BGP data and confirm that no other attribute towards the collector, e.g., the MED, has changed and no other prefix is included in the updates. However, since our observations are limited to inter-AS changes, we do not exclude the possibility for other sources of *nn* announcements, e.g. streams of updates due to intra-AS changes, misconfiguration, or rate limiting – in large and complex networks, the interplay of these multiple internal factors may lead to unnecessary updates.

**Revealed Information.** We have shown that geo-tagging can lead to bursts of announcements just updating the community attribute, which can lead to re-announcements by neighboring ASes, in a lab experiment and in the wild. Given the information hiding character of BGP, we next investigate how community exploration impacts the amount of information that is revealed by community attributes. We utilize the fixed announcement and withdrawal phases of the beacon prefixes, and label all announcements

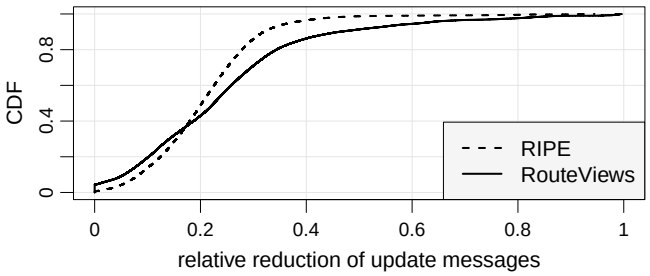


**Figure 7: Update message reduction of all MRT update files used in this work, across all time and all collectors.**

$\in d_{beacon}$ according to their appearances in any of the predefined phases, or outside them. We consider all announcements that appear withing 15 minutes of the respective phase begins, e.g., between 2:00 to 2:15 UTC for the first withdrawal phase.

In March 15, 2020, we identify a total of 21,398 unique community attributes. 62% of all community attributes are revealed exclusively during the withdrawal phases. Only 17% are revealed during the announcement phases and <1% outside both phases. The remaining attributes show up ambiguously. Historically, this distribution is stable, as can be seen in Figure 6. While the number of unique community attributes per day during withdrawal phases increased multifold in the last ten years, so did the total number, resulting in a stable ratio of about 60%. A reason for this high number is the global increase in connectivity between and within ASes and thus more alternative routes are explored during the withdrawal intervals. We note that tagged prefix aggregates are not reachable during global withdrawals.

**Update message reduction.** Another aspect to consider about unnecessary announcements is their impact on message archival. In order to investigate the update message overhead caused by unnecessary announcements, we return to our source data, i.e., all (251,493) update MRT files from which we have generated the data sets $d_{hist}$, $d_{mar20}$ and $d_{beacon}$. For each individual update file, we parse all the updates and check if they contain unnecessary announcements. Note that a single BGP update can contain multiple prefix announcements, as well as withdrawals for the same AS path. Given a single update file, we discard any update that contains only unnecessary announcements, e.g., no path changes or withdrawals. Since it is difficult to attribute a single prefix to the update message size, we conservatively count the full update message, even if it includes some unnecessary announcements.

Figure 7 shows the relative reduction of update messages over all MRT update files. We note that the distribution does not depend on the time or the collector. Still, since RouteViews and RIPE use a different a binning (96 and 288 update files per day, respectively) to archive the updates, here we distinguish between them. Around 50% of all RIPE update files are reduced by 20%+ update messages. While RouteViews show a slightly higher reduction effect (mean=24.8%), the overall reduction is comparable to RIPE (mean=20.8%). We note that the uncompressed file size correlates positively with the number of updates message it includes. Their ratio is almost constant for all update files per day, per collector (with a $\sigma$ of less than 0.05). Thus, we conclude that by removing updates containing only

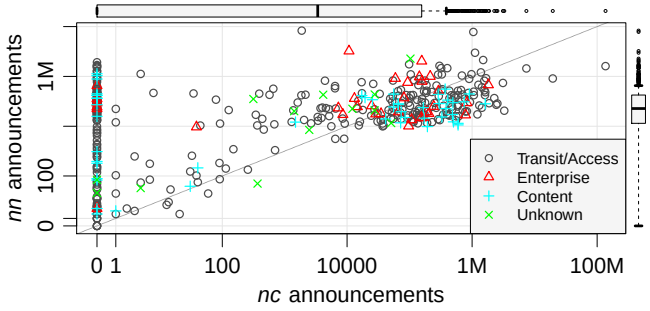


**Figure 8: Unnecessary announcement (*nc* vs. *nn*) behavior of peer ASes in $d_{mar20}$. No correlation with AS type.**

unnecessary announcements, the uncompressed data volume of update files can be reduced by 20-25%.

**Peer AS characterization.** Next, we take a closer look at ASes that send unnecessary updates, i.e., *nn* and *nc* announcements. Thereby, we focus only on ASes peering with collectors, since they provide the only ground-truth information in our data set. We note that not all peer ASes send updates on a given day. From the 581 peers in $d_{mar20}$, we select 451 that actually send updates to collectors. Figure 8 shows the number of *nc* announcements (x-axis) and *nn* announcements (y-axis) for those peers (both axes in log scale). We observe that while 162 (36%) peers send *nn* announcements only (x=0), the majority of peers (285 / 63%) send both, *nc* and *nn* announcements. Only 4 peer ASes send no unnecessary announcements (x and y=0). Interestingly, the cluster between 10K and 1M indicates a duality in the behavior of ASes: there are peers that send 10K+ of *nn* announcements only (x=0), and there are peers that send 10K+ of both, *nn* and *nc* announcements. In fact, there is not a single peer that sends *nc* announcements only, implying that when communities are forwarded duplicates are always involved. We believe this is the result of different community propagation and filtering behavior of the peer ASes.

We further assign an AS type to each of the peers by utilizing CAIDAs as2types data set [5]: We identify 355 Transit/Access, 49 Content, and 34 Enterprise ASes; for 13 peer ASes no type is available. We see that the different AS types spread across both dimensions. Thus, we conclude that the AS type does not correlate with the propagation and filtering behavior of ASes.

## 7 DISCUSSION

As the Internet's core inter-domain routing protocol, the BGP has been extensively studied. While previous work has found duplicate updates in the wild [24] and identified potential causes [21, 30], we show that BGP communities play a large role in the generation of unnecessary updates. First, as a transitive property of BGP messages, communities can induce updates to propagate through the entire routing system even when the path information is unchanged, the routing decision algorithm is unaffected, and the receiving AS does not recognize the community. Second, even when communities are filtered by an intermediate AS, common implementations still generate a duplicate update, just without the community. While duplicate updates without communities do not continue to propagate, we show that they represent a sizable fraction of BGP messages seen at route collectors and are unnecessary traffic.

Due to the surprising scale of our findings, we were interested to know if networkers and developers are aware of the behavior we observe and whether they consider it undesired. As part of our validation process, we created a website documenting our research efforts, including case studies, laboratory experiments, and open questions directed to the community [7]. We published this website on various mailing lists (NANOG [27], BIRD [3], FRRouting [12], and OpenBGPD [42]) and got into contact with various developers of routing daemons and two vendors. We have also directly contacted two operators of ASes involved in the propagation of unnecessary announcements, one of which, a large Tier-1 ISP, responded and confirmed one of our findings that only geolocation communities are forwarded to, e.g., customer ASes, while other communities are filtered. While the reaction of the community was affirmative of our findings, we highlight two perspectives on the generation of unnecessary updates. First, developers care about default configurations and to what degree they should predefine the behavior of a router. Indeed, as we show in laboratory experiments (§3) the default behavior deviates among different implementations. Also, through internal E-Mail communication we learn that even among the same vendor different teams of developers work separately on different implementations. Second, network operators need to weigh off between memory usage due to state keeping and the CPU overhead of processing unnecessary update messages. We find that a system wide increase of processing due to communities has not been anticipated by the community.

Prior work has shown that the lack of filtering and widespread propagation of BGP communities can leak information about networks' operation and practices [14, 16] and peering [15, 17], and can even be exploited to attack the routing system [41]. Our findings in this work demonstrate an additional motivation for more rigorous community filtering: reducing unnecessary duplicate update traffic. Not only does the unnecessary traffic impact router load and convergence times [1], it increases the load and storage requirements of systems that monitor BGP traffic including route collectors. We show that by removing unnecessary announcements from update messages archived by RIPE and RouteViews at least 20% of data volume can be saved, (§6). As the global use of communities increases and ASes become increasingly interconnected, the impact of not filtering will place even more strain on the system.

However, we note several other implications of our findings that we plan to study in future work. First, communities are somewhat paradoxical to BGP's emphasis on scalability and information hiding. For instance, the updates we observe often allow us to remotely infer the number of interconnections between two ASes and the location where they peer. Second, from observing updates and lack of updates at multiple points in the network, we can make rough guesses as to the way different ASes handle communities. Using more sophisticated network tomography techniques, we plan to classify per-AS community behavior, for instance those that tag, filter, and ignore. We hope to use this information to estimate the contribution of ASes that do not peer with collectors to the generation of unnecessary announcements. Finally, we believe that communities can enrich our understanding of anomalous behavior in the routing system beyond existing approaches. By characterizing the way ASes observe and process communities, our work provides a first step toward predicting anomalous communities.

## ACKNOWLEDGMENTS

We want to express our gratitude for the support and feedback received from the networking community via mailing lists and private communication, in particular Randy Bush, Greg Hankins, Jakob Heitz, Maria Matějka, Donald Sharp, Henk Smit and Stefan Wahl. We thank the reviewers and anonymous shepherd for their constructive input and guideance. This work supported in part by NSF grant CNS-1855614, the European Research Council (ERC) Starting Grant ResolutioNet (ERC-StG-679158), by the German Ministry for Education and Research (BMBF) as BIFOLD - Berlin Institute for the Foundations of Learning and Data (01IS18025A, 01IS18037A), and performed while the first author held an NRC Research Associateship award at the Naval Postgraduate School. Views and conclusions are those of the authors and should not be interpreted as representing the official policies or position of the U.S. government, the NSF, ERC, or BMBF.

## REFERENCES

[1] Sharad Agarwal, Chen-Nee Chuah, Supratik Bhattacharyya, and Christophe Diot. 2004. Impact of BGP Dynamics on Router CPU Utilization. In *PAM*. https://doi.org/10.1007/978-3-540-24668-8_28

[2] BIRD. 2020. The BIRD Internet Routing Daemon Project. https://bird.network.cz/. [Last accessed: October 20, 2020].

[3] Bird-users. 2020. A study on community-triggered updates in BGP. http://trubka.network.cz/pipermail/bird-users/2020-October/014922.html. [Last accessed: October 24, 2020].

[4] Larry Blunk, Craig Labovitz, and Manish Karir. 2011. Multi-Threaded Routing Toolkit (MRT) Routing Information Export Format. RFC 6396. https://rfc-editor.org/rfc/rfc6396.txt

[5] CAIDA. 2020. The CAIDA UCSD AS Classification - March 2020. https://www.caida.org/data/as-classification/.

[6] CISCO. 2005. Remotely Triggered Black Hole Filtering - Destination Based and Source Based. Cisco White Paper, http://www.cisco.com/c/dam/en_us/about/security/intelligence/blackhole.pdf.

[7] Community Exploration. 2020. A study on community trigged BGP updates. https://www.cmand.org/communityexploration/. [Last accessed: October 24, 2020].

[8] Shivani Deshpande and Biplab Sikdar. 2004. On the impact of route processing and MRAI timers on BGP convergence times. In *IEEE GLOBECOM*. https://doi.org/10.1109/GLOCOM.2004.1378136

[9] Christoph Dietzel, Anja Feldmann, and Thomas King. 2016. Blackholing at IXPs: On the Effectiveness of DDoS Mitigation in the Wild. In *PAM*. https://doi.org/10.1007/978-3-319-30505-9_24

[10] Benoit Donnet and Olivier Bonaventure. 2008. On BGP Communities. In *ACM SIGCOMM CCR*. https://doi.org/10.1145/1355734.1355743

[11] Anja Feldmann, Olaf Maennel, Z. Morley Mao, Arthur Berger, and Bruce Maggs. 2004. Locating Internet Routing Instabilities. In *ACM SIGCOMM CCR*. https://doi.org/10.1145/1015467.1015491

[12] frog – FRR Operator Group (users list). 2020. A study on community-triggered updates in BGP. https://lists.frrouting.org/pipermail/frog/2020-October/000980.html. [Last accessed: October 24, 2020].

[13] FRRouting. 2020. https://frrouting.org. [Last accessed: October 20, 2020].

[14] Vasileios Giotsas, Christoph Dietzel, Georgios Smaragdakis, Anja Feldmann, Arthur Berger, and Emile Aben. 2017. Detecting Peering Infrastructure Outages in the Wild. In *ACM SIGCOMM*. https://doi.org/10.1145/3098822.3098855

[15] Vasileios Giotsas, Matthew Luckie, Bradley Huffaker, and k claffy. 2014. Inferring Complex AS Relationships. In *ACM IMC*. https://doi.org/10.1145/2663716.2663743

[16] Vasileios Giotsas, Georgios Smaragdakis, Christoph Dietzel, Philipp Richter, Anja Feldmann, and Arthur Berger. 2017. Inferring BGP Blackholing Activity in the Internet. In *ACM IMC*. https://doi.org/10.1145/3131365.3131379

[17] Vasileios Giotsas, Georgios Smaragdakis, Bradley Huffaker, Matthew Luckie, and k claffy. 2015. Mapping Peering Interconnections to a Facility. In *ACM CoNEXT*. https://doi.org/10.1145/2716281.2836122

[18] Vasileios Giotsas and Shi Zhou. 2013. Improving the Discovery of IXP Peering Links through Passive BGP Measurements. In *IEEE INFOCOM WKSHPS*. https://doi.org/10.1109/INFCOMW.2013.6562878

[19] Vasileios Giotsas, Shi Zhou, Matthew Luckie, and kc claffy. 2013. Inferring Multilateral Peering. In *ACM CoNEXT*. https://doi.org/10.1145/2535372.2535390

[20] Timothy G Griffin and Gordon Wilfong. 1999. An Analysis of BGP Convergence Properties. In *ACM SIGCOMM CCR*. https://doi.org/10.1145/316194.316231

[21] David Hauweele, Bruno Quoitin, Cristel Pelsser, and Randy Bush. 2018. What do parrots and BGP routers have in common?. In *ACM SIGCOMM CCR*. https://doi.org/10.1145/3243157.3243159

[22] Jakob Heitz, Job Snijders, Keyur Patel, Ignas Bagdonas, and Nick Hilliard. 2017. BGP Large Communities Attribute. RFC 8092. https://rfc-editor.org/rfc/rfc8092.txt

[23] Thomas King, Christoph Dietzel, Job Snijders, Gert Doering, and Greg Hankins. 2016. BLACKHOLE Community. RFC 7999. https://rfc-editor.org/rfc/rfc7999.txt

[24] Craig Labovitz, G Robert Malan, and Farnam Jahanian. 1998. Internet routing instability. In *IEEE/ACM ToN*. https://doi.org/10.1109/90.731185

[25] Jun Li, Michael Guidero, Zhen Wu, Eric Purpus, and Toby Ehrenkranz. 2007. BGP Routing Dynamics Revisited. In *ACM SIGCOMM CCR*. https://doi.org/10.1145/1232919.1232921

[26] Tony Li, Ravi Chandra, and Paul S. Traina. 1996. BGP Communities Attribute. RFC 1997. https://rfc-editor.org/rfc/rfc1997.txt

[27] NANOG Mailing List. 2020. A study on community-triggered updates in BGP. https://mailman.nanog.org/pipermail/nanog/2020-October/210151.html. [Last accessed: October 24, 2020].

[28] Z. Morley Mao, Randy Bush, Timothy G. Griffin, and Matthew Roughan. 2003. BGP Beacons. In *ACM IMC*. https://doi.org/10.1145/948205.948207

[29] OpenBGPD. 2020. http://www.openbgpd.org/. [Last accessed: October 20, 2020].

[30] Jong Han Park, Dan Jen, Mohit Lad, Shane Amante, Danny McPherson, and Lixia Zhang. 2010. Investigating Occurrence of Duplicate Updates in BGP Announcements. In *PAM*. https://doi.org/10.1007/978-3-642-12334-4_2

[31] Vern Paxson. 1996. End-to-End Routing Behavior in the Internet. In *ACM SIGCOMM*. https://doi.org/10.1145/248156.248160

[32] Quagga. 2020. Quagga Software Routing Suite. https://www.quagga.net/. [Last accessed: October 20, 2020].

[33] Yakov Rekhter, Susan Hares, and Tony Li. 2006. A Border Gateway Protocol 4 (BGP-4). RFC 4271. https://rfc-editor.org/rfc/rfc4271.txt

[34] Philipp Richter, Georgios Smaragdakis, Anja Feldmann, Nikolaos Chatzis, Jan Boettger, and Walter Willinger. 2014. Peering at Peerings: On the Role of IXP Route Servers. In *ACM IMC*. https://doi.org/10.1145/2663716.2663757

[35] RIPE. 2020. Current RIS Routing Beacons. https://www.ripe.net/analyse/internet-measurements/routing-information-service-ris/current-ris-routing-beacons. [Last accessed: June 2, 2020].

[36] RIPE. 2020. RIS - RIPE Network Coordination Centre. http://ris.ripe.net/. [Last accessed: June 2, 2020].

[37] Matthew Roughan, Walter Willinger, Olaf Maennel, Debbie Perouli, and Randy Bush. 2011. 10 Lessons from 10 Years of Measuring and Modeling the Internet's Autonomous Systems. In *IEEE J-SAC*. https://doi.org/10.1109/JSAC.2011.111006

[38] RouteViews. 2020. University of Oregon RouteViews project. http://www.routeviews.org/. [Last accessed: June 2, 2020].

[39] Bijal Sanghani. 2015. Euro-IX Update. https://www.euro-ix.net/media/filer_public/d7/64/d764cdcc-efeb-42a1-8ef1-85239b283bca/euroix-update-1115.pdf.

[40] Job Snijders, John Heasley, and Martijn Schmidt. 2017. Use of BGP Large Communities. RFC 8195. https://rfc-editor.org/rfc/rfc8195.txt

[41] Florian Streibelt, Franziska Lichtblau, Robert Beverly, Anja Feldmann, Cristel Pelsser, Georgios Smaragdakis, and Randy Bush. 2018. BGP Communities: Even more Worms in the Routing Can. In *ACM IMC*. https://doi.org/10.1145/3278532.3278557

[42] users@openbgpd.org. 2020. A study on community-triggered updates in BGP. https://marc.info/?l=openbgpd-users&m=160305012627575. [Last accessed: October 24, 2020].

[43] Feng Wang, Zhuoqing Morley Mao, Jia Wang, Lixin Gao, and Randy Bush. 2006. A Measurement Study on the Impact of Routing Events on End-to-End Internet Path Performance. In *ACM SIGCOMM CCR*. https://doi.org/10.1145/1151659.1159956