Internet Science Moonshot: Expanding BGP Data Horizons

Thomas Alfroy[1], Thomas Holterbach[1], Thomas Krenc[2], KC Claffy[2], Cristel Pelsser[3]
[1] University of Strasbourg, [2] CAIDA, [3] UCLouvain

ABSTRACT

Dramatic growth in Internet connectivity poses a challenge for the resource-constrained data collection efforts that support scientific and operational analysis of interdomain routing. Inspired by tradeoffs made in other disciplines, we explore a fundamental reconceptualization to how we design public BGP data collection architectures: an *overshoot-and-discard* approach that can accommodate an order of magnitude increase in vantage points by discarding redundant data shortly after its collection. As defining *redundant* depends on the context, we design algorithms that filter redundant updates without optimizing for one objective, and evaluate our approach in terms of detecting two noteworthy phenomena using BGP data: AS-topology mapping and hijacks. Our approach can generalize to other types of Internet data (e.g., traceroute, traffic). We offer this study as a first step to a potentially new area of Internet measurement research.

CCS CONCEPTS

• **Networks** → **Network measurement**;

KEYWORDS

Internet measurement, BGP, Routing Security

ACM Reference Format:
Thomas Alfroy[1], Thomas Holterbach[1], Thomas Krenc[2], KC Claffy[2], Cristel Pelsser[3]. 2023. Internet Science Moonshot: Expanding BGP Data Horizons . In *The 22nd ACM Workshop on Hot Topics in Networks (HotNets '23), November 28–29, 2023, Cambridge, MA, USA*. ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3626111.3628202

Acknowledgements: This work was supported by the ArtIC project (grant ANR-20-THIA-0006-01), Région Grand Est, Inria Nancy-Grand Est, IHU of Strasbourg, U. of Strasbourg, U. of Haute-Alsace, the Internet Society, NSF CNS-2120399 and NSF OAC-2131987. Views are those of the authors and do not represent the endorsements of the funding agencies.

1 INTRODUCTION

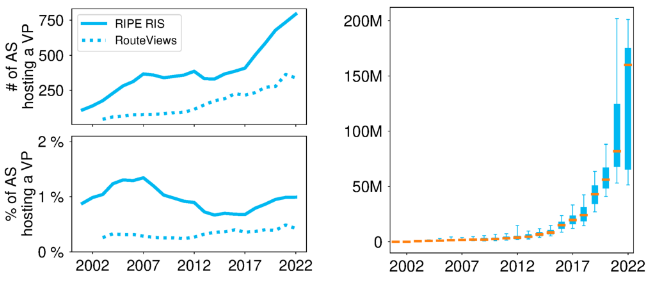
The BGP routing data published by two global projects over the last two decades—RIS [14] and RouteViews [13]—fuels an entire field of Internet infrastructure research, as well as many systems that measure and monitor Internet routing. Each project operates BGP collectors that peer with operational Internet routers around the world and collect their routes. As this data has become increasingly critical, RIS and RouteViews have made efforts to *(i)* deploy more Vantage Points (VPs), i.e., peer with more routers that export their routes to the collection service (see Fig. 1a, top), and *(ii)* provide all the collected routing data to users, via either real time provisioning systems or archived files on disk.

Three realities in today's landscape merit re-evaluating how we think about Internet routing data collection. First, RIS and RouteViews, operated by nonprofit organizations, are constrained by funding and resources which limits their ability to expand their peering footprint and thus their visibility of Internet routes. Each project peers with a few hundred ASes—a small fraction of the ASes [5] (Fig. 1a, bottom)—yet they already struggle with the volume of data they collect [1]. These annual storage requirements of these two projects collectively grew from 33TB in 2020 to 78TB in 2023.

Second, the deployment of new VPs amplifies the data storage requirements caused by the growth of the Internet itself: the number of unique IP prefixes (e.g., due to de-aggregation or new assignments) constantly grows [5], as well as the number of unique ASes and links between them. Thus, even with a constant number of VPs, the volume of routing data inevitably increases, contributing to a quadratic increase of observed updates over time (Fig. 1b).

Third, persistent challenges with deploying routing security protections (e.g., against route leaks and hijacks), and growing concerns from governments about slow progress in this area [18], has highlighted the importance of these collector projects as primary source for detecting both accidental and malicious transgressions in the routing system. However, given the information-hiding character of BGP, the current collection strategies do not provide comprehensive visibility of the global routing system. Researchers have recently demonstrated how strategically-scoped attacks can evade visibility of current collection systems [12].

We explore data collection methods to accommodate a radical increase, e.g., by an order of magnitude, in the number of VPs feeding public collection systems. While significant investment in data collection could accommodate gathering, retention, and sharing orders of magnitude more routing data, current constraints require a more strategic approach



(a) Growth in VPs over time. **(b) # of updates per hour.**

Figure 1: RIS and RouteViews continuously expand over time. The number of routes that they collect per hour jumped from 82M in 2021 to 160M in 2022.

to gathering and retaining BGP data. In this paper, we explore novel approaches to *retaining* the data.

Overshoot and discard. Our vision is to fundamentally change the way we collect BGP data, adopting a new *overshoot-and-discard* strategy. Akin to CERN's Large Hadron Collider (LHC) which generates millions of collisions just to see a few interesting particles (e.g., Higgs boson), overshooting BGP data collection will maximize the chance to see interesting routing events, e.g., BGP hijacks. We imagine a world where public BGP data providers could automate deployment of additional VPs, targeting a *moonshot* of peering with one VP in every of the ≈75K ASes participating in the global routing system (even half would be a moonshot!). Overshooting BGP data collection is only feasible if the system can discard the "less interesting" bits upon acquisition, before it consumes processing or storage resources. In the case of the LHC, fast online algorithms using custom hardware and software discard 99.994% of the likely less interesting collisions [16].

Predictability of BGP data streams. Predictable and redundant properties of BGP data streams [2, 4] suggest that BGP data may be amenable to filtering with minimal loss of information. For example, often several VPs observe BGP routes with similar—sometimes even identical—attribute values. Although there is some signal in knowing which VPs observed the same prefix, we propose to explore the practical implications of filtering data with substantial redundancy. Consider the goal of BGP hijack detection. A hijack may reach many VPs, in which case we will have redundant visibility of it, or may reach no available VP (perhaps by intention [12]), in which case substantial expansion of VP deployment is the best way to increase the chance of observing it.

Challenges. The key question is which BGP updates to discard as missing data inevitably implies loss of information and impacts some studies more than others. This question is hard to answer as updates are used for various purposes and there is no consensus on which performance metrics to use and on how to define useful and redundant updates.

Executing our vision also raises new fundamental research questions such as: how to update the decisions about which data to discard over time, how to discard routes with minimal infrastructure changes, and how can we ensure that discarding routes does not open new attack vectors.

First successes. We undertake a preliminary exploration of this vision. We sketch the outline of a system, GillNet, that data providers such as RIS and RouteViews could install to collect BGP routes in an overshoot-and-discard manner. We use a probabilistic prediction framework to show that BGP routes are highly predictable and filtering them carefully leads to minimal loss of information. With GillNet, data providers can accommodate more VPs (e.g., remote peering sessions) by configuring *"route-maps"* in the collectors to filter redundant updates.

We use GillNet to evaluate the impact of collecting routes using an overshoot-and-discard approach when focusing on a subset of the currently existing VPs. With the same volume of data collected, GillNet allows the same analysis approach to reveal 35% more hijacks and 84% more AS links compared to when processing all the data. We expect the benefit of GillNet to increase as more VPs are deployed.

2 MOTIVATING ANALYSIS

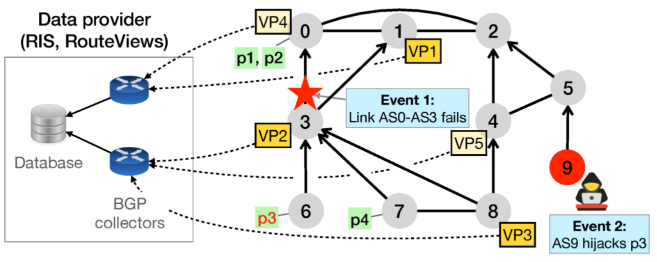
We illustrate the gain of deploying more but filtered VPs over the current deployment strategy. Fig. 2a shows ten ASes (0-9) inter-connected in customer-to-provider (arrows) and peer-to-peer (lines) relationships, according to the Gao-Rexford model [9]. *AS0* originates two prefixes p1 and p2 whereas *AS6* and *AS7* each originate one prefix, p3 and p4, respectively. Among the ASes, five VPs (1-5) peer with BGP collectors (dashed lines) maintained by data providers like RIS or RouteViews. We consider two isolated events: (1) an Internet outage (link cut at the physical layer) impacting the peering session between *AS0* and *AS3*, and (2) a hijack where *AS9* illegitimately announces p3, the prefix owned by *AS6*. We only consider the updates induced by these two events.

Current approach. Assume a deployment of only three VPs: VP1, VP2, and VP3. In this case, the route collectors observe four updates (Fig. 2b), induced exclusively by a link failure (event 1) causing a path change via 1. Although the current approach discards no updates, the three-VP deployment reveals a link failure in one direction but not the other, limiting the inference of the nature of the link failure. More importantly, while the hijack (event 2) induces updates, they go unnoticed by the three VPs because of their shorter AS path toward the legitimate owner, 6. This hijack is undetectable in this scenario.

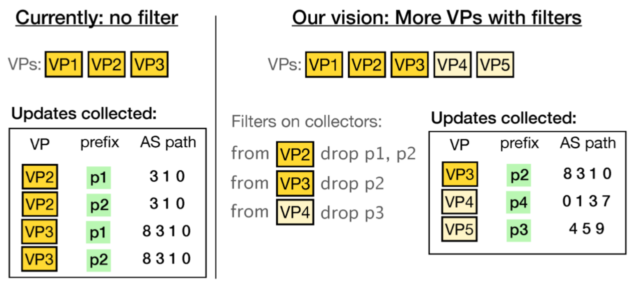
With the overshoot-and-discard data collection strategy. Assume all five VPs are deployed with the following filters, which aim to discard redundant updates, configured on the BGP collectors (filter generation details in §3):

```
from VP2 drop announcements for prefixes p1, p2;
from VP3 drop announcements for prefix p2;
from VP4 drop announcements for prefix p3.
```

With these filters, the route collectors observe three BGP updates. Yet, these three updates allow more useful inferences



(a) A scenario describing an AS-level topology where a few VPs send their routes to a data provider. We focus on two events: An outage on link 0 - 3 and a hijack by *AS9* on p3.


Currently: no filter

VPs: VP1 VP2 VP3

Updates collected:

VP	prefix	AS path
VP2	p1	3 1 0
VP2	p2	3 1 0
VP3	p1	8 3 1 0
VP3	p2	8 3 1 0

Our vision: More VPs with filters

VPs: VP1 VP2 VP3 VP4 VP5

Filters on collectors:

from VP2 drop p1, p2
from VP3 drop p2
from VP4 drop p3

Updates collected:

VP	prefix	AS path
VP3	p2	8 3 1 0
VP4	p4	0 1 3 7
VP5	p3	4 5 9

(b) Our overshoot-and-discard approach (right) collects fewer updates from more useful VPs, which better reveals routing events compared to current approach (left).

Figure 2: A scenario highlighting why our overshoot-and-discard approach is beneficial when collecting BGP data.

than the original four updates. In fact, the update received by VP3 enables detection that one direction of the link 0—3 is not used, and the update received by VP4 (for p4 with path 0—1—3—7) enables detection that the other direction is not used, which is useful to infer the failure (e.g., with [8]). Moreover, VP5 is close to the hijacker and observes the hijacked route, which is preferred over the legitimate ones in this region of the topology. Once collected by the platform, this hijacked route enables monitoring systems to detect the hijack and report it to the victim.

Key takeaways. This scenario demonstrates the possibility of gathering more insight from less but intelligently filtered BGP data. We purposely placed additional VPs and optimized filters to detect the two routing events and discard updates with similar attribute values (e.g., the four updates pertaining to p1 and p2 and observed by VP2 and VP3 have a similar AS path and only one is retained). In practice, there is no ground truth about which routing events will appear, where, and which updates to filter. To make the overshoot-and-discard approach practical, we propose to design a system (§4) that automatically builds the filters based on correlations between BGP updates in historical data.

3 EXPLORING CORRELATIONS

We build a framework that automatically captures correlations between collected BGP updates without optimizing for a particular objective. These correlations let us infer whether we can reconstitute a BGP update from other updates—in which case discarding it minimizes loss of information.

Correlation graphs. We use correlation graphs to probabilistically capture correlations between BGP updates. We build one correlation graph for every distinct prefix observed. We denote $\mathcal{G}(p) = (V_p, E_p)$ the correlation graph for prefix p, with V_p the set of nodes and $E_p \in V_p * V_p$ the set of directed edges. Each edge in E_p has a weight in $\mathbb{R}^+$. A correlation graph differs from the formal definition of a graph as it connects *sets* of updates. More precisely, a node $N = (v, o, a, c) \in V_p$ is the set of updates for prefix p received by the VP v, with the origin AS o, the AS path a and the community values c. N can include more than one update, in which case these updates have identical attribute values but different timestamps. The weight of the directed edge going from node $N_x = (v_x, o_x, a_x, c_x)$ to node $N_y = (v_y, o_y, a_y, c_y)$ is the *correlation ratio.*

DEFINITION 1 (CORRELATION RATIO). *The correlation ratio $C(N_x, N_y)$ is the proportion of updates in N_x that correlate with at least one update in N_y.*

Observe that $C(N_x, N_y)$ and $C(N_y, N_x)$ can be different. The correlation ratio is used to build the edges: there is an edge from N_x to N_y only if $C(N_x, N_y)$ is greater than zero. Thus, the graph might not be connected. We define the correlation between two updates as follows:

DEFINITION 2 (CORRELATION BETWEEN TWO UPDATES). *An update $u_1 \in N_x$ observed at time t_1 correlates with update $u_2 \in N_y$ observed at time t_2 if these two conditions hold:*

$$|t_1 - t_2| < 2 \textit{ minutes and } o_x = o_y$$

The first condition adds a slack to accommodate typical BGP convergence time [11]. The second condition ensures that two updates with similar attribute values but a different origin AS do not correlate. This condition ensures that any update induced by an origin hijack does not correlate with legitimate updates.

Example. We use the scenario in Fig. 2 to show how our correlation graphs capture correlations in BGP data. We consider the BGP updates induced by the following sequence of four events, each of which is separated by more than two minutes.

T1: Link 0—3 is down;
T2: Link 0—3 is up;
T3: Links 0—3 and 3—8 are down simultaneously;
T4: Links 0—3 and 3—8 are up simultaneously.

Fig. 3 shows how we build the correlation graph for p1 (recall that we build one correlation graph for *every* prefix). *Upon T1:* VP2 observes that p1 is reachable via 3—1—0, which creates node N_1. VP3 observes that p1 is reachable via 8—3—1—0, which creates node N_2. These two updates are correlated as they are triggered by the same event (Def. 2), thus $C(N_1, N_2) = 1$ and $C(N_2, N_1) = 1$.

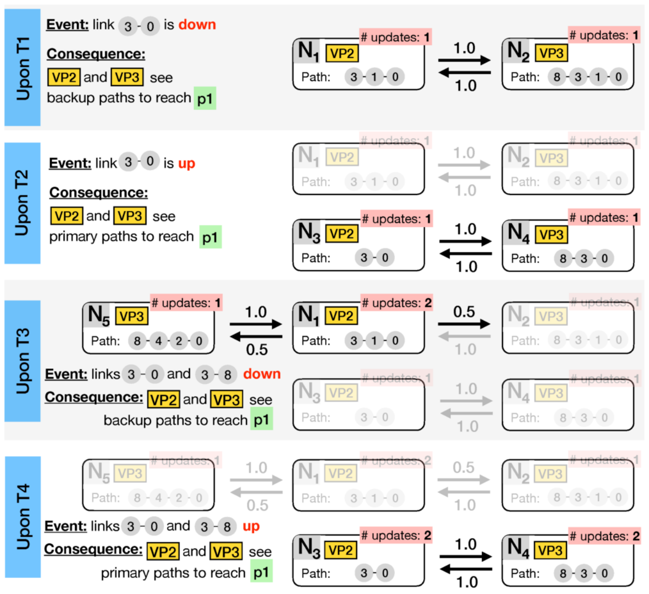

Upon T1
Event: link 3 - 0 is down
Consequence: VP2 and VP3 see backup paths to reach p1
Upon T2
Event: link 3 - 0 is up
Consequence: VP2 and VP3 see primary paths to reach p1
Upon T3
Event: links 3 - 0 and 3 - 8 down
Consequence: VP2 and VP3 see backup paths to reach p1
Upon T4
Event: links 3 - 0 and 3 - 8 up
Consequence: VP2 and VP3 see primary paths to reach p1
# updates: 1
# updates: 2
Path: 3 - 1 - 0
Path: 8 - 3 - 1 - 0
Path: 3 - 0
Path: 8 - 3 - 0
Path: 8 - 4 - 2 - 0
1.0
0.5

Figure 3: Constructing the correlation graph for p1 and four events triggered at time T1, T2, T3 and T4. We omit community values (irrelevant here) and origin AS (which is always *AS0*) in nodes. Each node shows (in pink) the number of updates it contains.

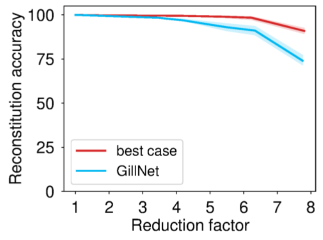
Upon T2: VP2 and VP3 observe that the best paths to reach p1 are functional again, which creates two nodes N_3 and N_4 with $C(N_3, N_4) = C(N_4, N_3) = 1$. The correlation ratio between nodes N_1, N_2 and N_3, N_4 is zero as the time difference between T1 and T2 is greater than two minutes.

Upon T3: VP2 observes that p1 is reachable via 3—1—0. The update is added in N_1, which now includes two updates, one observed at T1 and the other at T3. Consequently, we have $C(N_1, N_2) = 0.5$. VP3 observes that p1 is reachable via 8—4—2—0, which creates node N_5. We have $C(N_5, N_1) = 1$ whereas $C(N_1, N_5) = 0.5$ as N_1 includes two updates but only one correlates with the update in N_5.

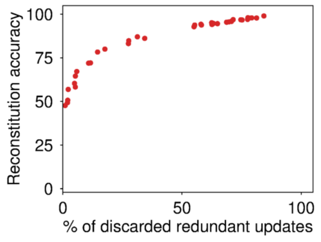
Upon T4: VP2 and VP3 observe that the best paths to reach p1 are functional again. The updates are added in N_3 and N_4, respectively. However, the correlation ratios between N_3 and N_4 do not change and are still equal to one.

Ability to reconstitute original data. To explore how to filter BGP updates with minimal information loss, we develop an algorithm that splits an input set of BGP updates into two subsets α and β, and tries to reconstitute β from updates in α. Our algorithm builds the correlation graph using 24 hours of BGP updates. Then, for each update in α, it finds the node in the correlation graph that shares identical attribute values and iterates over its successors, i.e., nodes with which it has a positive correlation ratio. Finally, the algorithm reconstitutes the updates in the visited nodes.

Example. Assume the correlation graph in Fig. 3 and consider that VP3 observes at time t (with $t > T4$) an update for



(a) We can reconstitute filtered updates from retained ones as they are correlated.



(b) Discarding redundant updates is key to high reconstitution accuracy.

Figure 4: Experimental analyses reveals that a set of BGP updates is highly reconstitutable when when redundant updates are removed.

p1 with path 8—3—1—0 and origin *AS0*. The reconstitution algorithm adds this update into N_2 and reconstitutes the updates in N_1, the only successor of N_2.

Performance metrics. We evaluate our algorithm with two metrics: reconstitution accuracy and reduction factor.

Reconstitution accuracy: The proportion of updates in β that our algorithm identically reconstitutes (true positive rate), including timestamp (with a two-minute slack as in Def. 2). Observe that our reconstitution algorithm yields an upper bound of the true positive rate (i.e., a best case) at the cost of reconstituting more nonexistent updates, i.e., false positives.

Reduction factor: Ratio between number of updates in the original dataset and number of updates input to the reconstitution algorithm: $(|\alpha| + |\beta|)/|\alpha|$.

Finding #1: We can effectively reconstitute BGP updates. We run our reconstitution algorithm for all prefixes, on 20 distinct ten-minute intervals that do not overlap with the 24-hour period used to build the correlation graph. We tested different *reduction factors*, ranging from 1 to 8. Fig. 4a shows the *reconstitution accuracy* as a function of the reduction factor. The red line is the median accuracy over the 20 runs, and the lightly colored area around the line indicates the best and worst accuracy. Our reconstitution algorithm can reconstitute 98.9% of the original updates (median) from only 20% of them. This result reflects the high correlation across BGP updates, and shows that it is possible to discard a large portion of them with minimal loss of information.

Finding #2: The key factor to obtain a high reconstitution accuracy is to discard redundant BGP updates. We take a set of updates from which we discard a proportion, leaving set α (with a *reduction factor* of six), and use our algorithm to reconstitute the discarded updates, i.e., β. We repeat this experiment 40 times for different sets of updates and vary (between 20 and 100) the percentage of discarded updates that are redundant with at least one retained update (in α), i.e., the percentage of redundancy removed from the original dataset. We define two updates as redundant if they have the same prefix, same origin AS, and if their time difference is below two minutes. Fig. 4b shows that the higher the fraction of redundant updates discarded, the more accurately we can reconstitute the original dataset. Further, for a fixed proportion of discarded redundant updates, the *reconstitution accuracy* varies slightly, which demonstrates that discarding redundant updates is key to minimzing information loss.

4 GILLNET

We present GillNet, a system that takes as input a set of VPs and some historical BGP data and builds filters that can be configured on BGP collectors (e.g., using *route-maps*) so that data providers can process and store only the less redundant updates. The best-case algorithm used in §3 to find which updates to discard is not applicable in GillNet for two reasons. First, GillNet operates in real-time, without ground truth. Second, while filters could match on any BGP attribute (e.g., prefix, AS path or community), we restrict GillNet to filter updates based on their prefix only to reduce the search space and facilitate the evolution of filters over time (see §6).

GillNet's algorithm to build filters. GillNet relies on correlation graphs that it builds for all prefixes using 24 hours of data. GillNet inspects the correlation graphs and finds updates that have little or no correlation with other updates. These updates are in nodes that have a low sum of incoming edge weights (N_2 or N_5 in Fig. 3). Intuitively, these updates carry routing information that is hard—or even impossible—to reconstitute from other updates. GillNet prioritizes accepting these updates over the more predictable and redundant ones, which aligns with our findings in §3.

More formally, consider the correlation graph $\mathcal{G}(p)$ for prefix p. We denote $w(\mathcal{G}(p), N, N')$ the weight (i.e., correlation ratio) of the link from N to N' in $\mathcal{G}(p)$. We denote $P(\mathcal{G}(p), N)$ the set of N's predecessors. We define the *global correlation score* $\mathcal{S}(\mathcal{G}(p), N)$ of node N in $\mathcal{G}(p)$ as:

$$\mathcal{S}(\mathcal{G}(p), N) = \sum_n^{P(\mathcal{G}(p),N)} w(\mathcal{G}(p), N, n)$$

GillNet picks nodes with a low *global correlation score* and prioritizes their updates, as they are harder to predict from the other updates. Then, GillNet builds the filters to apply on the configured BGP sessions between collectors and VPs. However, GillNet can either filter all the updates for a given prefix and VP, or none of them. In Fig. 3, GillNet cannot build filters to accept updates in N_5 and discard the ones in N_2 as they are collected by the same VP (VP3). Thus, for every prefix, GillNet sorts the VPs based on their *average global correlation score* $\overline{\mathcal{S}}$, which we define for VP v and prefix p as:

$$\overline{\mathcal{S}}(p, v) = \frac{\sum_n^{nodes(\mathcal{G}(p),v)} \mathcal{S}(\mathcal{G}(p),n)}{|nodes(\mathcal{G}(p),v)|}$$

with $nodes(\mathcal{G}(p), v)$ the set of nodes in $\mathcal{G}(p)$ that include updates observed by VP v. In Fig. 3, the *average global correlation score* for VP2 and VP3 are computed as:

$$\overline{\mathcal{S}}(p, \boxed{\text{VP2}}) = \frac{\mathcal{S}(\mathcal{G}(p),N_1)+\mathcal{S}(\mathcal{G}(p),N_3)}{2} = \frac{2+1}{2}$$

$$\overline{\mathcal{S}}(p, \boxed{\text{VP3}}) = \frac{\mathcal{S}(\mathcal{G}(p),N_2)+\mathcal{S}(\mathcal{G}(p),N_3)+\mathcal{S}(\mathcal{G}(p),N_4)}{3} = \frac{0.5+0.5+1}{3}$$

Finally, GillNet generates filters to accept updates for a prefix from the k VPs with the lowest *average global correlation score* for that prefix, and to discard updates from others, with k depending on the number of updates to discard.

GillNet's data reconstitution power. We evaluate the data reconstitution power of GillNet using our reconstitution algorithm described in §3. We now aim to reduce false positives to make GillNet practical. Given an update in α and its corresponding node N in the correlation graph, GillNet's reconstitution algorithm visits node N' (a neighbor of N) and reconstitutes its corresponding update with a probability p, where p is the weight of the edge connecting N and N'. We run this algorithm on 20 ten-minute periods that do not overlap with the 24-hour period used to build the correlation graph and with reduction factors ranging from one to eight.

GillNet only returns a slightly lower TPR than the best-case algorithm (Fig. 4a): it can reconstitute 94.5% of the original dataset (median) with a reduction factor of five. GillNet inevitably constructs updates that are not present in the original dataset β. This is the case for 56% of the constructed updates (in the median case and with reduction factor of five), which is low compared to the number of nonexistent updates that could be reconstructed (true negatives).

5 ESTIMATED GAINS AND LOSSES

We use GillNet to estimate how an overshoot-and-discard strategy enables extensive data analysis without requiring BGP data providers to process and archive more data. We estimate the gains and losses induced by discarding data, and show that the tradeoff is largely in favor of GillNet's overshoot-and-discard strategy.

Methodology. It is impossible to run GillNet on an imaginary Internet where every AS deploys one VP and evaluate its impact on every possible objective. We thus estimate the impact of GillNet when run on a set of 500 existing RIS and RouteViews VPs, and focus on the following two popular and orthogonal (in scope) objectives.

Hijack detection: The goal is to detect origin hijacks, by detecting whether the origin AS in a BGP route is authorized to announce the prefix.

AS topology mapping: Our goal is to discover as many AS links as possible. We parse the AS path observed in every BGP update, discard the AS paths including a private AS number, and take the remaining AS links.

We compare the efficiency of sets of BGP updates to achieve the two objectives when collected using two data collection schemes: one that we name NoFilter (i.e., collecting all updates as RIS and RouteViews do today) and one using GillNet. RIS and RouteViews compress the BGP data, stored in MRT format [3], using either *gzip* or *bzip2*. We ensure that the comparison is fair and consistent with the route collector storage strategies by considering datasets of identical size, after transforming the updates into MRT format and compressing the MRT files using *bzip2*.

The gains. Fig. 5 shows the number of hijacks (left) and AS links (right) revealed from the data collected with GillNet and with NoFilter. We tested GillNet with reduction factors

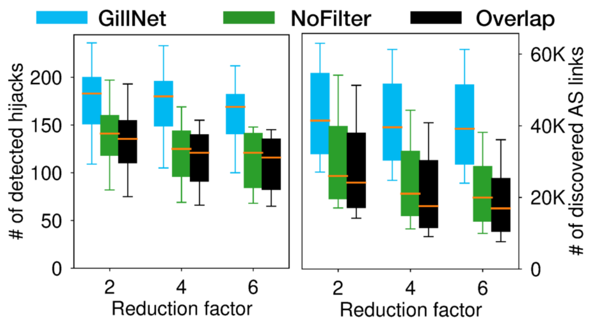


Figure 5: Comparison of the efficiency of sets of the BGP updates to reveal hijacks (left) and AS links (right) when collecting them using GillNet or NoFilter. Whiskers represent 5th and 95th percentiles; the red line is the median.

of two, four, and six. For every reduction factor, we perform 30 experiments using data collected from different randomly selected sets of 500 VPs. For both objectives and every reduction factor, GillNet collects more useful updates. With a reduction factor of 4, the updates collected by GillNet enable detection of 180 hijacks against only 125 when collecting updates using NoFilter, and 39566 AS links versus 21034. A promising aspect of the overshoot-and-discard approach is that the quality of GillNet-based inferences does not significantly drop when doubling or tripling the reduction factor. This result suggests that *(i)* GillNet successfully discards redundant updates and *(ii)* the level of redundancy in the BGP data is so high that we can discard a large portion and use the saved resources to deploy more VPs.

The losses. Fig. 5 shows the *overlap* in number of hijacks and AS links revealed with both strategies. This number is close to the number of hijacks and AS links revealed using NoFilter, which indicates that the data discarded by GillNet results in a limited loss of information.

6 OPEN CHALLENGES

We discuss open challenges and future directions in exploring, adapting, and operationalizing our proposed approach.

Building and operating a next generation BGP data collection platform. While RIS and RouteViews could quickly benefit from GillNet by using *"route-maps"* to configure GillNet-provided filters, scaling up by an order of magnitude opens several design and operational challenges.

Hardware and software scalability: Handling >10K sessions can yield extreme bursts of updates, combined with managing potentially billions of filters, will increase memory and CPU demands. We envision either a distributed version of the collector, which balances sessions across servers, and/or accelerated packet processing using e.g., eBPF or SmartNICs.

Automation: Scaling an order of magnitude will require automated configuration of new BGP peers (or BMP [17]). Minimizing the risk of fake or misconfigured peering sessions will require automated security checks, e.g., validating information against PeeringDB, as bgp.tools does today [6].

Incentivization. A non-technical obstacle to scaling the platforms is providing many ASes with incentive to peer with a collector. Selling BGP analytics in exchange for a peering session is one approach, e.g., many peers of the bgp.tools system opt in to share their BGP feed with researchers. Other transparency and accountability efforts are emerging that may provide such incentives in the future [7, 10].

Keeping accurate filters over time. Many factors (e.g., deployment of a new VP) require updating filters on BGP collectors, which is challenging because the calculus for discarding requires as input *all* routes observed in a previous window. One approach is to rely on transient "sentinel" filters that accept all routes for a given prefix such that the system can build a historical dataset for that prefix and update its filters. This approach requires exploring tradeoffs, since any control-plane update consumes resources on the collector. GillNet's design simplifies this process, by iteratively updating the filters prefix per prefix (i.e., independently).

Preventing new attack vectors. In its current state, we expect that GillNet opens two attack vectors.

Adversarial inputs: GillNet relies on past data to build filters, so it is subject to adversarial inputs allowing an attacker to manipulate future filters.

Evading public collectors: We expect users will know which filters are in use. Attackers can exploit this information to try to engineer their malicious announcements into the discarded data, as in [12]. Researchers could analyze the feasibility of such attacks, and design mitigation schemes. Platforms operators could also leverage the fact that many possible sets of filters will discard redundant routes, and consider switching among them in ways that make it harder to predict which traffic will be discarded.

Finding consensus. Discarding data will impact some use cases more than others. The question of which data to discard requires community consideration and discussion. We can envision a hybrid approach where data providers process the full feed of a few carefully selected VPs and apply filters on others. Another direction is to investigate for how many, and for which VPs, data providers should process all the data.

Developing lossless compression approaches. Current BGP collection systems have operated for two decades with little change to the data architectures, e.g., formats, cadence of full RIB dumps, etc. Our redundancy detection framework could inspire approaches that encode this redundancy in a lossless way. The tradeoff would be a data format that requires decoding to use and new tool chains to support it.

Beyond BGP. Our proposed overshoot-and-discard approach extends to other types of BGP monitoring systems (e.g., BMP) and other types of Internet data, e.g., active measurement platforms (e.g., RIPE Atlas [15]). We hope our proposal opens up a new and exciting area of Internet research, inspired by other disciplines that have had to make difficult tradeoffs in pursuit of a more complete understanding of the universe.

REFERENCES

[1] Emile Aben. 2020. Route Collection at the RIPE NCC - Where are we and where should we go? https://labs.ripe.net/author/emileaben/route-collection-at-the-ripe-ncc-where-are-we-and-where-should-we-go/.

[2] Dionysus Blazakis, Manish Karir, and John S Baras. 2006. BGP-Inspect-extracting information from raw BGP data. In *IEEE/IFIP Network Operations and Management Symposium NOMS*.

[3] Larry Blunk, Craig Labovitz, and Manish Karir. 2011. Multi-Threaded Routing Toolkit (MRT) Routing Information Export Format. In *RFC 6396*.

[4] Kai Chen, Chengchen Hu, Wenwen Zhang, Yan Chen, and Bin Liu. 2009. On the Eyeshots of BGP Vantage Points. In *GLOBECOM*.

[5] CIDR. 2023. CIDR REPORT. https://www.cidr-report.org/as2.0/.

[6] Ben Cox. 2023. BGP tools. https://bgp.tools/.

[7] David Clark, Cecilia Testart, Matthew Luckie, kc claffy. 2023. A path forward: preventing path hijacks by enabling trust zones. https://catalog.caida.org/paper/2023_a_path_forward.

[8] Anja Feldmann, Olaf Maennel, Z. Morley Mao, Arthur Berger, and Bruce Maggs. 2004. Locating Internet Routing Instabilities. *SIGCOMM*.

[9] Lixin Gao and Jennifer Rexford. 2000. Stable Internet Routing without Global Coordination. In *SIGMETRICS*.

[10] Internet Society. 2022. Enhancing MANRS (Mutually Agreed Norms for Routing Security). https://www.manrs.org/2022/09/improving-manrs-call-for-working-group-participation/.

[11] Craig Labovitz, Abha Ahuja, Abhijit Bose, and Farnam Jahanian. 2000. Delayed Internet Routing Convergence. *SIGCOMM CCR*.

[12] Alexandros Milolidakis, Tobias Bühler, Kunyu Wang, Marco Chiesa, Laurent Vanbever, and Stefano Vissicchio. 2023. On the Effectiveness of BGP Hijackers That Evade Public Route Collectors. *IEEE Access*.

[13] University of Oregon/Network Startup Resource Center. 2021. Route Views Project. www.routeviews.org/.

[14] RIPE. 1999. RIS Raw Data. https://www.ripe.net/data-tools/stats/ris/.

[15] RIPE. 2010. The RIPE Atlas measurement platform. https://atlas.ripe.net/.

[16] Sciencealert. 2018. Less Than 1% of Large Hadron Collider Data Ever Gets Looked at. https://www.sciencealert.com/over-99-percent-of-large-hadron-collider-particle-collision-data-is-lost.

[17] John Scudder, Rex Fernando, and Stephen Stuart. 2016. BGP Monitoring Protocol (BMP). RFC 7854 *(Request for Comments)*. https://www.rfc-editor.org/info/rfc7854

[18] U.S. Federal Communications Commission. 2022. NOTICE OF INQUIRY. PS Docket No. 22-90. In the Matter of Secure Internet Routing. https://www.fcc.gov/ecfs/document/1022806680214/1.