Detecting Traffic Engineering from public BGP data

Omar Darwich, Cristel Pelsser, Kevin Vermeulen

Detecting Traffic Engineering from public BGP data

Omar Darwich[1], Cristel Pelsser[2], and Kevin Vermeulen[1]

[1] LAAS-CNRS
[2] Université de Louvain

Abstract

Routing is essential to the Internet functioning. However, more and more functions are added to BGP, the inter-AS routing protocol. In addition to providing connectivity for best effort service, it carries flow specification rules and blackholing signals to react to DDoS, routes for virtual private networks, IGP link-state database information among other uses. One such addition is the tweaking of BGP advertisements to engineer the traffic, to direct it on some preferred paths. In this paper we aim to estimate the impact of Traffic Engineering (TE) on the BGP ecosystem. We develop a method to detect the impact in space, that is, to find which traffic engineering technique impacts which prefix and which AS. We design a methodology to pinpoint TE events to quantify the impact on time. We find that on average, a BGP vantage point sees 35% of the announced prefixes impacted by TE. Quantifying the impact of TE on BGP stability, we find that TE events contribute to 39% of BGP updates and 44% of the BGP convergence time, and that prefixes belonging to hypergiants contribute the most to TE.

1 Introduction

The Internet is composed of a large number of independent networks that are interconnected, called Autonomous Systems (AS). They exchange routing information using the Border Gateway Protocol (BGP). This routing protocol enables Internet Service Providers (ISP) to configure routing policies meeting their economic relationships. While economical relationships take precedence and are implemented using the first attribute considered in the route selection, the local_pref, operators may wish finer control on their traffic. BGP provides a set of tools to engineer the incoming and outgoing traffic of operators.

Traffic Engineering (TE) can be practiced at different timescales. It is either done manually or using tools such as Noction [6], BGP-TE-TOOL [5] and AutoPrepend [18]. Each change of routing configuration is an open door for potential miss-configuration [38] such as routing leaks, unintentional hijacks [19], routing loops, forwarding loops, and disconnections. Some popular outages are the result of routing configuration changes [2].

Despite the existence of TE mechanisms in BGP, their granularity is very coarse [25, 43], especially for incoming traffic. Further, each service provider

tries to achieve its own goals that may be in contradiction with other operators' needs. For instance, the TE systems from Google and Meta [45, 51] aim to direct outgoing traffic to specific routers when the customer edge network may prefer a different provider for traffic from Google and Meta. Knowing the risk of performing changes in BGP and the little or sometimes inexistent results of performing TE [14], one can wonder whether it is worth the effort, and understand its impact on the BGP ecosystem.

In this paper, to bridge this gap, we perform the first large-scale study of the prevalence of the different TE techniques and how TE impacts BGP stability.

Our contributions are:

- New techniques to detect TE from BGP dumps and BGP updates, informed from operators and principled on BGP behavior.
- A longitudinal study on the impact of TE on affected prefixes and ASes, as well as the impact of TE on BGP stability.

Our main results include that our technique is highly accurate. Our technique has a precision of 0.996 precision and recall of 0.91 recall on our ground truth datasets. Using our technique to detect traffic engineering over the years, we find that TE has been and is widely used by operators. In 2024, a BGP vantage point sees an average number of prefixes affected by TE of 35%. We also find that TE has a significant impact on BGP stability finding that TE contributes to 39% of the BGP updates and 44% of the BGP convergence time of the BGP events[3].

2 Background and related work

First, we highlight the important elements of the Border Gateway Protocol (BGP) for our work. Then, we explain what is Traffic Engineering (TE), why it is used, and how it works from an interdomain point of view.

Networks or Autonomous Systems (ASes) exchange reachability information via BGP. The purpose of the protocol is for each AS to learn how to reach all the prefixes allocated on the Internet. A route to a prefix in BGP contains multiple attributes, including the sequence of AS numbers that need to be traveled. This sequence is called an AS Path. Interdomain TE is done by carefully crafting and manipulating BGP announcements to steer traffic as desired [42].

Quoitin *et al.* distinguish incoming and outgoing TE [42], where the goal is to influence how the traffic reaches an AS and how it leaves an AS, respectively. In this paper, we focus on incoming TE. The outgoing techniques are out of scope of this paper, but are presented in Appendix B for completeness.

2.1 Inbound TE

There are four classes of inbound traffic engineering techniques. We illustrate these techniques using Figure 1.

[3] BGP events represent a subset of the total BGP updates, as we filter out continuous BGP updates that are likely to not be real BGP events (§3.2).

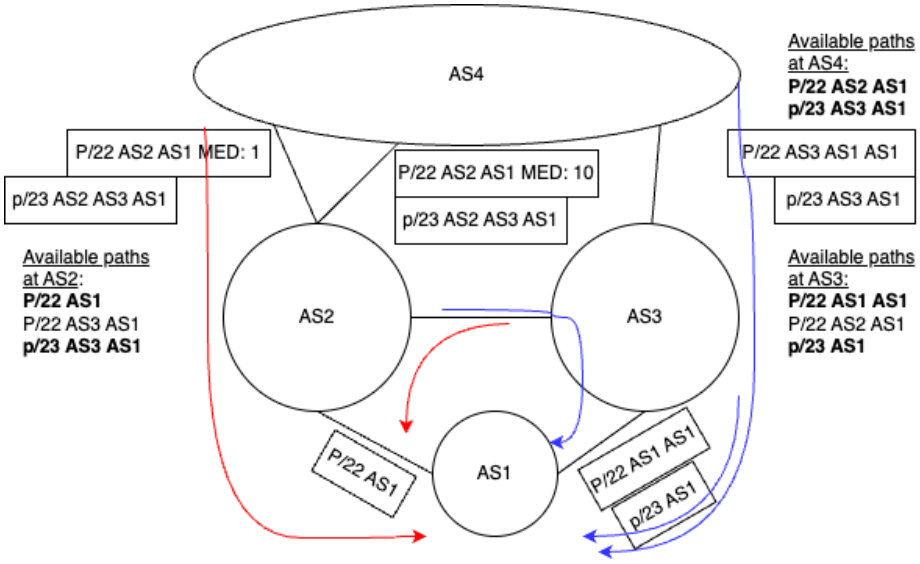

Fig. 1: Overview of incoming TE techniques: $p/23$ is a more specific prefix of $P/22$. $P/22AS1AS1$ is an advertisement of $P/22$ with AS path prepending. $AS2$ advertises a low MED on the left link and a higher MED on the right link.

Selective advertisements: Routes for different prefixes are treated independently. The best announcement is selected for each prefix. ISPs may split their IP address space in sub-prefixes. The sub-prefixes are selectively advertised to some peers and not to others. We use the term peer to mean any other AS where the AS announcing the prefix has a BGP session with. To ensure reachability even in the case of connectivity failure to one peer, a super-prefix covering the sub-prefixes is advertised to all peers. This route is only used when a sub-prefix route disappears. Because traffic is forwarded using the route for the most specific prefix, the selective advertisements of the sub-prefixes, when present, enable control of the incoming paths used for the different sub-prefixes.

In Figure 1, $P/22$ is the super-prefix and $P/23$ one of its two sub-prefixes. $AS1$ advertises $P/22$ to both of its providers ($AS2$ and $AS3$). $p/23$ is solely advertised to $AS3$ to attract the traffic destined to the IP addresses in this prefix on link $AS3 - AS1$ instead via $AS2$. The paths selected for the sub-prefix $p/23$ by the different ASs are shown in blue. The paths to $P/22$ are in red. If link $AS3 - AS1$ fails. $p/23$ disappears from the routing tables in the different ASs. The IP addresses in $p/23$ are still reachable following the routes for $P/22$.

AS path prepending: The BGP decision process is composed of a set of rules that are applied in sequence [17]. One of these rules is to keep only the routes with the shortest AS paths. The assumption is that shorter AS paths means a slower delay. To indicate to a remote AS that a path is not preferred, an AS can add its AS number more than once to the AS path. This technique is

called AS prepending. Since the AS path becomes longer, all things remaining equal, the path becomes less preferred than shorter paths.

In Figure 1, $AS1$ prepends its AS path once when advertising $P/22$ to $AS3$. $AS3$ receives the path $AS1AS1$ from $AS1$ when it receives the path $AS2AS1$ from AS2. Since both have the same length, it chooses one of them based on the next rules. Here we assume it picks the path via $AS2$. Here all the traffic to $P/22$ except when destined to sub-prefix $p/23$ goes via $AS2$. The TE action enables to move all the traffic on link $AS2 - AS1$.

BGP communities: BGP communities are a set of values attached to a BGP route. Each community value has semantics that is defined by each AS. A community value is attached to a BGP route by a router to be acted upon later by another router on the route. A community value may convey an action such as AS path prepending, not exporting the route to a peer, or to a set of peers, or setting the local-preference of a route to a value. A community value may be added to a route to instruct another AS to influence route advertisement and thus perform a TE action. These community values are called BGP action communities. Other BGP communities, which carry information such as where the route was learnt, are called information communities.

In Figure 1, instead of prepending the AS path in $AS1$ toward $AS3$, $AS1$ could instruct $AS3$ to perform the prepending for $P/22$. In that case, we would most likely see the AS path $AS3AS3AS1$ instead of $AS3AS1AS1$. Alternatively, $AS1$ could attach the "no-export'' community toward $AS3$ for $P/22$. In both cases, we observe the same routes.

Multi-exit discriminator: Multiply connected neighboring ASes may indicate which link they prefer the neighbor to send the traffic on using the Multi-Exit Discriminator (MED) attribute. A low MED value is inserted in the advertisements on the preferred links while a higher value is assigned on the less preferred ones. This attribute does not propagate further. It is rarely observed in public route collectors. We may remotely observe the local effect of the MED if location communities are also present in the announcements. It may also trigger some duplicate announcements [32]. Otherwise this TE technique will be invisible to publicly available route collectors.

In Figure 1, a MED is attached by $AS2$ to the updates for prefix $P/22$ when advertised to $AS4$. The lowest MED route wins. Consequently, traffic uses the left link from $AS4$ to $AS2$ for the super-prefix.

Since TE with the MED attribute is invisible to public collectors, we focus on the 3 other techniques in the rest of the paper.

2.2 Related work

Several works propose methods to help network operators engineer inbound TE. For instance, Chang *et al.* propose to automate AS path prepending [18], when Sun *et al.* rely on more specific prefix advertisements to engineer inbound TE for stub ASes [50]. Nakamura *et al.* show that outbound traffic can be influenced using BGP EPE to pick the exit points and reach paths with shorter RTTs than the delay experienced along the default best paths [40].

On the other hand, few works address the detection of TE, as a whole, and rather focus on a specific question for a given TE technique. In [31], Gutierrez proposes to detect AS path prepending from BGP updates with varying lengths but constant nominal AS path (the AS path without duplicate ASes). This technique misses the events that lead to a switch of path due to TE. Prior work also looked at how path prepending was used by operators and the different implications in terms of security and performance [39]. On BGP communities, prior work proposed to classify them [36, 37, 48], as well as studying their impact on security [35, 49]. Our work is orthogonal to these works as it gives a landscape of the usage of these different TE techniques, comparatively to each other.

Related to detecting TE events from BGP updates (§3.2), prior work has analyzed BGP updates to detect typos [22], BGP sequences [13] or classify the updates based on the attribute changes [33]. Our work proposes new principled techniques to detect signals of TE from the BGP updates.

3 Detecting traffic engineering

Our goal is to capture the impact of TE in space and time on the BGP ecosystem. By space, we mean finding which prefixes and which ASes are impacted by TE, and by which techniques. By time, we mean measuring the role of TE in BGP instability. To find which prefixes are affected by TE, we translate the TE techniques described in Section 2.1 into observable properties from BGP dumps (§3.1). To measure the role of TE in BGP instability, we design a methodology to capture TE events from BGP updates based on insights on TE practices and how BGP should behave for TE and non TE events (§3.2).

3.1 Detecting TE from BGP dumps

The BGP dumps from the BGP collectors contain, among other information, the AS path and the BGP communities used to reach prefixes in the Internet, which we use to reason about and find prefixes affected by TE. We refer to BGP collectors as vantage points (VP) throughout the rest of this document. For each technique, we explain how we derive whether a prefix is affected by this technique, and how we derive if the origin AS is responsible for the TE, if we can.

Selective advertisement Our idea is that if we observe a prefix and a covering prefix with the same origin AS with different routes from the same vantage point, it is likely that there is an AS on the path performing selective advertisement. Indeed, if the routes to the two prefixes were announced exactly in the same way, then their path should be the same. However, in detail, it might be tricky to definitely identify selective advertisements. Indeed, suppose we have these two paths: A-B-C-O the path to the sub-prefix, A-B-E-O the path to the super-prefix, where O is the origin AS announcing the two prefixes. These paths could be the result of O performing selective advertisement, only announcing the sub-prefix to C, or B which sets a higher local preference to C for the sub-prefix, and

a higher local preference to E for the super-prefix. To help differentiate, the view from multiple providers can help. Indeed, if from multiple providers, we see other paths like G-F-C-O and G-F-E-O, it is more likely to be selective advertisements than F and E applying exactly the same policies for the sub-prefix and the super-prefix. Therefore, if C and E have more than one common neighbor observed for the sub-prefix and the super-prefix (here B and F), we say there is selective advertisement on the sub-prefix and the super-prefix.

Notice that our technique allows us to differentiate between selective advertisement and an AS performing deaggregation of all its prefixes, for instance to prevent BGP hijacks. Indeed, in the case of deaggregation, the paths of the selective prefix and the super-prefix should be the same, whereas it is not the case in selective advertisement.

It can be tricky to identify the AS which wants to perform selective advertisement. In our previous example, the selective advertisement is performed by O as the AS paths of the sub-prefix and the super-prefix diverge directly after O. However, suppose that the paths to the sub-prefix and the super-prefix are now A-B-C-G-O and A-B-E-G-O. In that case, the selective advertisement could be performed by G, or by O. Indeed, O could use a community of G asking G to only announce the selective sub-prefix to B. Therefore, to be conservative, we only consider the cases where we can be definitive in saying that the selective advertisement is performed by the origin AS, *i.e.,* when the paths to the sub-prefix and the super-prefix directly diverge after the origin AS. These cases represent 81% of the prefixes with selective advertisement (§4.3).

AS path prepending If we observe an AS path from a vantage point with an AS appearing multiple times subsequently, we say that the AS path contains prepending. We distinguish between origin prepending and intermediate prepending, depending on whether the origin AS is the prepended AS or not.

For origin AS prepending, the AS performing the TE must be the origin AS. For intermediate prepending, it might be harder to identify who is responsible, as it could be any AS between the prepended AS and the origin AS, so to not overestimate which ASes perform TE, we do not consider that intermediate prepending is made by the origin AS.

AS path poisoning If we observe an AS path containing the pattern A-*-A, where the "star" is a sequence of ASes without A, we say that the AS path contains poisoning, and that A is the origin AS that performs the poisoning. When observing poisoning, the AS appearing multiple times in the AS-path did not receive the update a second time. Otherwise, it would have dropped the update, and we would not have seen it in the public data. It is also possible for a router to disable loop detection, which is typically used for ASes that lack a backbone and rely on another AS for internal connectivity [20]. These paths are often not seen at public collectors as the AS disabling the check is a stub. Third, these AS paths could also be due to BGP hijacks [47], and therefore the origin AS is not responsible for the TE action. In practice, we quantify the occurrences

of AS path poisoning but we do not investigate their cause as they represent a negligible fraction of affected prefixes ($< 0.01\%$) (§5.2).

BGP communities To find communities, we use the methodology from prior work that classified the BGP communities between action and information [37], which obtained an accuracy of 96% on a dataset containing the BGP communities seen in a week of BGP data from May 2023. More specifically, we replicate prior work's technique [37], using the available code [34], on each week of our BGP updates (§5.1). Then, each prefix with a BGP community classified as action is said to be affected by TE. We conservatively do not consider prefix updates with the information community as being affected by TE. Moreover, we do not differentiate between the case where the action community was actually taken into account by the operator, and the case where the action community was ignored by the operator. For both, we say that the prefix was affected by TE, as we are interested in knowing how operators are performing TE, and not whether their operation is successful. This is similar to prepending being performed by an operator, but potentially ignored because BGP tie breakers with higher priority (*e.g.,* localpref) are used for path selection.

To identify whether the origin AS is responsible for the action community, we look for an action community of the form `penultimate_asn:xxx` where penultimate_asn is the AS just before the origin AS in the AS path. If such a community exists, then it is very likely that the action community has been set by the origin AS. However, this technique will give a lower bound on the set of origin ASes that performs traffic engineering, for the following reasons: (1) Prior work has shown that action communities might be off path, *i.e.,* the action community appear only on AS paths where the AS that has to perform the TE action is not on the AS path [37] (2) Some providers have action communities that do not start with their AS number. For instance, NTT has some communities starting with 65XXX [11], whereas their AS number is 2914 (3) An origin AS can try to influence the BGP announcement more than one hop away.

Finally, as communities are not verified on the path and can be set by anyone [49], our technique would not work if an AS (other than the origin AS) polluted an announcement with an action community of the form `penultimate_asn:xxx` (erroneously or maliciously).

We evaluate the limitations of our technique to detect the origin AS setting an action community in Section 4.3.

Other techniques The other techniques, namely setting different values of local preference, and the usage of the MED attribute are not visible at BGP vantage points. Their use can be inferred in the close vicinity of a VP only as shown by Cittadini et al. [21] for the local preference. Therefore, we do not try to find the prefixes affected by these TE techniques. Consequently, our study provides a lower bound of the quantity of traffic engineering performed on the internet.

3.2 Detecting TE events from BGP updates

Intuition: The methodology described in the previous section gives us a way to compute the set of prefixes on which TE techniques are applied, but does not give us a way to capture TE events, as BGP dumps only represent a single point in time. In this section, we design a methodology to capture TE events, based on two principled insights derived from time and space behavior of operators performing TE.

First, it is unlikely that operators perform TE on all their prefixes at the same time, and conversely, non TE events, such as link or device failures, are likely to affect all prefixes that are routed similarly at the same time. Tools like Traffic Manager [8] exemplify this approach by selectively applying TE to a subset of prefixes. We briefly give an overview of Traffic Manager later in this section. Indeed, there are multiple reasons for which operators would avoid performing TE on all their prefixes, among which we can cite: (1) Any configuration change represents a risk [52], and (2) Operators using TE to find potential alternate paths, either for load balancing or to mitigate congestion, will use only a fraction of prefixes to test the alternate paths [45].

Second, it is likely that if we observe multiple stable paths within a short period of time, an operator is performing TE, for instance for searching alternate paths, or testing failure scenarios [52]. Indeed, multiple stable paths are unlikely to be caused by outages, as most outages are short lived [41] and should not result in observing multiple stable paths, but rather either a very few stable paths corresponding to the state before the failure, the state during the failure, and the state after the restoration of the failure. Also, multiple stable paths are unlikely to happen from configuration errors preventing BGP to converge, as one would observe a lot of updates resulting in multiple unstable paths.

A TE system from a major CDN confirmed our intuitions [8]. This TE system is in charge of automatically shifting traffic from a datacenter to some others in case this datacenter was running hot. To achieve this goal, this TE system automatically drops some prefixes announced at a particular datacenter (but not all) to shift the desired amount of traffic. This mechanism confirms our first intuition that not all the prefixes experience TE at the same time. Then, another job of this TE system is to predict where the traffic would go in case of a failure of a peering session or even an entire datacenter. To achieve this prediction, the TE operation consists in withdrawing prefixes from the tested datacenter, waiting for convergence, and performing pings to the clients that were routed to that datacenter and observe to which other datacenter they are rerouted [23]. This operation is run continuously to keep the traffic prediction up to date, so it confirms our second intuition that some TE operations consist in testing failure scenarios that would translate into multiple stable paths.

We discussed with the CDN to confirm our understanding of their TE system, and the CDN also added that they could sometimes withdraw all the prefixes of one datacenter at once to simulate failures, which contradicts our first intuition. We further discuss this case in Section 6. As we design our technique to stick with our two intuitions (§3.2), it is likely that it would not be able to detect

those events. However, even if one could consider these events as TE as these are not real failures and operator's operations, these simulated failure scenarios confirm our intuition of what would happen during a real failure.

We describe how we translate these two insights into concrete techniques in the next paragraphs.

Parsing BGP events from BGP updates We say that a route to a prefix in BGP is stable if it lasts at least 15 minutes before receiving any other BGP update. This 15 minutes value is chosen for two reasons: (1) We want to ignore noisy prefixes with continuous updates [1, 24]; (2) We are interested in BGP events corresponding to TE, not all the events, and operators should not announce their routes more frequently than every 15 minutes, as they could be penalized by route flap damping [30]. With this definition, a route change event is a sequence of BGP updates observed from a BGP vantage point for a prefix such that the routes before and after the updates are stable and different. We will use the term BGP event for this concept in the rest of the paper.

This definition of BGP events gives that over our dataset (§5.1), only 11% of the BGP updates are part of a BGP event. This is not surprising, as prior work has demonstrated that a significant fraction of the BGP updates were not due to real path change [13, 32]. Some are due to the lack of adj-RIB-ins or presence of a partial adj-RIB-in in routers. Because the router does not store the routes it received, when a policy changes locally it needs to ask its neighbors to send their best route to compute its new best route. This is for example observed upon route validity change due to a ROA change. There is a current IETF draft that proposes a solution for the specific case of validity change [9].

Operators should rarely perform TE on all their prefixes simultaneously Our intuition, confirmed by a large CDN, is that a BGP event is likely due to TE only if a fraction in a group of prefixes that are routed similarly moves simultaneously. To translate this idea into a practical methodology, we use the notion of BGP atoms [12]. We use the same definition as in prior work, which defined a BGP atom as the set of prefixes that have the same AS path towards them, and so the BGP atoms are computed per vantage point.

To compute which BGP events are due to TE, for each BGP event, its associated prefix, called candidate TE prefix, and its vantage point, we compute the set of prefixes that were in the candidate TE prefix's BGP atom in a two-minute window before the starting time of the event, to account for BGP propagation [28]. Then, we compute the fraction of prefixes in the atom that also experience an AS path change seen from the same vantage point during the BGP event. If this fraction is lower than a threshold α, we label the BGP event as TE. The threshold implements our first intuition that not all prefixes of an AS are affected in one TE event.

In addition, if we see a subsequent BGP event after the TE event where the AS path changes back to the path before the TE event, we also label the subsequent BGP event as TE: Let us suppose that an AS O announces a prefix p

and our vantage point observes the path VP-A-O to reach p. Then, O performs TE on p so that we observe a BGP event where the path to reach p changes from VP-A-O to VP-B-O. To realize this change, O had to make the path VP-B-O more preferred than VP-A-O, using one of the techniques described in Section 2. If O uses selective advertisement or a community to not announce to A, or if O uses prepending, the AS path VP-A-O does not exist, so if we observe it back, it means that an additional TE action was performed to make it exist again. In theory, complex routing policies with multiple links between ASes and MED (§2.1) could invalidate our reasoning, but our heuristic works well in practice (§4.1).

To be clear, we only apply this technique when the first event is TE. Indeed, if the first event was not labeled as TE and was due, for instance, to a link failure, seeing the first AS path again could just indicate that it was restored.

Multiple stable paths are likely due to TE Our second idea is based on how BGP should behave during an outage. Say we observe a path change for the prefix p announced by AS O. It is observed from the vantage point at AS VP, and the path changed from VP-A-O to VP-B-O, because of an outage. When the outage ends, assuming that there was no other BGP event during the outage (and most outages on the Internet are short lived [41], so this should be a reasonable assumption), BGP will typically converge back to the path before the outage, or to another one because the tie breaker used to select the path was the time when the router learnt the path.

In other words, in case of an outage, one should observe at most three stable AS paths, namely the path before the outage, the path during the outage, and the path after the outage, making a fourth one unlikely. Therefore, if we observe four stable AS paths during a short period of time, we label the BGP events corresponding to the different AS path changes as TE.

One challenge with this heuristic is to find the right time window where a sequence of stable paths can correspond to TE. Indeed, the longer the window, the higher the coverage will be, at the cost of precision. From discussion with operators, TE operations should not last more than a few hours in general. Indeed, operators can perform TE to react to timely events such as traffic peaks or proactively shift traffic [8], but they can also perform planned maintenance. For this last point, we obtained from the operators of the PEERING testbed [46] to forward the emails from their neighbors informing PEERING that there will be maintenance. Over the period of August-September 2024, PEERING received three emails from two operators, where the maintenance time was between four and ten hours. We evaluate the tradeoff between precision and coverage in Section 4.1.

Summary A BGP event is labeled as TE if it respects one of the two conditions described in the previous sections. Our methodology has three parameters: (1) the threshold α of the maximum fraction of prefixes in the same atom that can change path together during a TE event, (2) the number of stable paths that

need to be observed over a short period of time for the events associated to be labeled as TE, and (3) the time window to use to find TE events based on the multiple stable paths. We evaluate our choice for these values in Section 4.1.

4 Evaluation

Overall, our technique has a precision of 0.996 with only 20 False Positives (FP) on our ground truth datasets, and a recall of 0.91 (§4.1). It also has a high "consistency" over events from multiple vantage points (§4.2). In addition, the parameters that we chose, namely the maximum fraction of prefixes in the same BGP atom simultaneously moving (0.5), the number of stable paths (4) and the duration to look for stable paths (4 hours) to label TE events provide the best precision without sacrificing recall. Finally, our heuristics to capture prefixes with TE from BGP dumps correctly identify them on our ground truth (§4.3).

4.1 Precision and recall

Datasets: To collect TE events, we use three different datasets: 1) a dataset from prior work using TE to mitigate DDoS attacks [44], called *DDoS TE dataset*, 2) a dataset from a project deployed on the PEERING testbed [46] using BGP action community to perform TE [4], called *PEERING action community TE dataset*, 3) and our own TE dataset, called *Own TE dataset*.

The DDoS TE is produced from a setup that primarily utilized two anycasted prefixes. One of these prefixes served as a control and remained unchanged throughout the experiments. For the second prefix, it was announced from five different sites with varying levels of prepending. Initially, no prepending was applied at any site. Gradually, one site at a time added prepending, ranging from 1 to 5 repetitions. The experiment also included a reverse prepending phase, where every site started with 5 repetitions of prepending on the prefix. One site at a time then reduced the prepending incrementally. This experimental design demonstrated how prepending could be employed in the context of an anycasted prefix to steer undesired traffic away from an affected anycast site during a DDoS event. We selected the data for the prepended prefix from the week of 2022/03/21.

The PEERING action community TE dataset comes from an ongoing research work directly obtained from the authors, that performs TE in the context of failover. The authors announce a prefix from a main site and multiple backup sites using extensions of different BGP announcement strategies to influence how the clients are routed in case of the failure of the main site. In particular, for our context, they use BGP action communities from their backup sites to restrict the set of providers to which they announce their prefix. We obtained this dataset directly from the authors, which contain the logs of their announcement with the timestamp. The experiments were run on 28/3/2024.

[4] This work is ongoing work and an extension from prior work [52].

Our own TE dataset contains four /24 prefixes belonging to a covering /22, that we announce from our AS during one week starting at 2023/07/10. Our AS has two upstream providers, and we perform selective advertisements for three of our /24 prefixes, changing the provider to which the prefix is announced every 3, 6, and 12 hours respectively. The last /24 prefix and the /22 do not change their announcement.

To collect non TE events, we mimic prior work [16] to retrieve outages, which says that there is an outage if both the IODA platform [10] and the Cloudflare Radar [7] platform detect an outage. We retrieve two outages: one in AS 34700 from 2024/01/11 13:50 to 2024/01/12 00:30 UTC, and one in AS 27839 from 2023/01/12 20:10 to 2023/01/12 20:40 UTC. We call this dataset Outage dataset.

For the DDOS TE, Action community TE and the Own TE datasets, we have the precise timestamps of the announcements and withdrawals corresponding to TE events, so we can retrieve the corresponding updates in publicly available BGP data. We collect BGP updates from the RIPE RIS RRC00 full feed collector peers for the corresponding periods of time. We retrieve a total of 854 TE events for DDoS TE dataset, 216 events from the PEERING action community TE dataset, 4,850 TE events for the Own TE dataset, and 2,714 non TE events in the Outage dataset. For the non TE event, we only consider the events happening during the 15 minutes after the start of the outage as being part of the outage, and thus a non TE event.

Results: We label TE events with a positive label, and non TE events with a negative label. Table 1 shows the recall on the three datasets with TE events and the precision on the forth dataset with non TE events. Our technique has a high precision (>0.99) with only 20 false positives. The recall is also high (0.91), but not perfect, with only 2 false negatives for the DDoS TE dataset, 536 for the Own TE dataset and 0 for the PEERING action community TE dataset.

We investigate further the false negatives to understand why our technique failed to correctly classify them. For the Own TE dataset, 175 events are due to the overlapping periodicity of our announcements: As the three prefixes exactly changed their announcements every 3, 6, and 12 hours, there are times where they all changed together while being in the same BGP atom, making our algorithm erroneously declare that these events were not due to TE. For the remaining 361 events, the prefix is alone in its atom and does not meet the criteria of the multiple paths, showing a limitation of our technique to detect TE.

For the DDoS TE dataset, we observe that prefixes are alone in their atoms before changing paths. The high performance of our technique on this dataset is caused by the numerous stable paths observed during a short period of time.

Grid search to find the right parameters to tune our technique: Table 2 shows the precision and recall when we vary the three parameters to label events as TE: (1) α, the threshold of the fraction of prefixes simultaneously moving in the same BGP atom, (2) the number of stable paths, and (3) the time period to observe those stable paths. We observe that values of α=0.5, 4 stable paths, and a time period of 4 hours give the best tradeoff on our evaluation datasets, with a

Dataset	TP	TN	FP	FN
DDoS TE [44] (2022)	852	N/A	N/A	2
PEERING action community TE (2024)	216	N/A	N/A	0
Own TE (2023)	4314	N/A	N/A	536
Outage [16] (2023 and 2024)	N/A	2694	20	N/A

Table 1: Performance of our TE detection technique on four different datasets.

precision of 0.996 and a recall of 0.91, so we select these values for the rest of the paper. In Section 5.4, we evaluate how this choice affects our result.

Other sources of events and negative results: To be totally transparent, we also report on our unsuccessful attempts to gather additional ground truth datasets. First, we tried to gather TE events directly from operators. We sent a survey to NANOG, FRNOG, and SWINOG, asking the operators if they would like to help us evaluate our technique by giving pairs of (timestamp, prefix) corresponding to their TE operations. We got responses from two operators, one recommending to change a question in the survey, and another one specifying that this operator was actually performing TE with prepending, but could not provide more specific information as they were not keeping this kind of information, so it was not usable for us to collect TE events.

In addition, we directly contacted four operators in August and September 2024, our R&E network, one major transit provider, one Japanese provider and one major CDN, in which we had privileged contacts, but we did not obtain any response from the operational teams at the time of the submission.

4.2 Consistency

We evaluate the consistency of our technique over multiple vantage points, with the intuition that if a TE event is seen by a vantage point, other vantage points seeing an event at the same time should also label it as TE.

We define consistency as follows: assuming we have a set of events, containing at least one TE event, we define the consistency over a group as the number of events labeled as TE divided by the number of events of this set. The sets of events are computed by grouping events close in time to the *TE event* together. For each TE event, we build a set of events from the detected events, from other vantage points, starting within a 2 minute window of the TE event. As we want to count each event only once, we perform a disjoint set algorithm on these sets of events to obtain our final sets.

Datasets: We collect BGP updates from the RIPE RIS RRC00 full feed collector peers for the week of January 8th 2024 to January 15th 2024. Applying our methodology to detect TE events and grouping them to compute consistency, we obtain 14,547,990 groups of events containing at least one TE event.

Results: The average consistency over the different groups of events of 0.98, with 96% with a consistency of 1, showing that most of the vantage points agree that an event is TE when at least one vantage point says so. We also verify that

α	# of Stable paths	Time period for considering stable paths (hours)	Precision	Recall
0.3	3	4	0.98	0.87
0.3	3	6	0.72	0.91
0.3	3	8	0.72	0.92
0.3	4	4	0.998	0.82
0.3	4	6	0.87	0.85
0.3	4	8	0.86	0.87
0.3	5	4	0.999	0.81
0.3	5	6	0.98	0.82
0.3	5	8	0.96	0.82
0.5	3	4	0.98	0.94
0.5	3	6	0.73	0.96
0.5	3	8	0.73	0.97
0.5	4	4	0.996	0.91
0.5	4	6	0.88	0.93
0.5	4	8	0.86	0.94
0.5	5	4	0.998	0.9
0.5	5	6	0.98	0.91
0.5	5	8	0.96	0.91
0.7	3	4	0.98	0.94
0.7	3	6	0.73	0.96
0.7	3	8	0.73	0.97
0.7	4	4	0.99	0.91
0.7	4	6	0.88	0.93
0.7	4	8	0.86	0.94
0.7	5	4	0.99	0.9
0.7	5	6	0.98	0.91
0.7	5	8	0.96	0.91

Table 2: Precision and recall when performing a grid search on the three parameters: (1) α, the maximum fraction of prefixes that can move simultaneously in the same atom, (2) the number of stable paths observed, and (3) the time window duration where we observe those paths.

there are indeed multiple vantage points that agree that there is a TE event when there is a TE event, and that we do not have a lot of cases where there is only one vantage point seeing the TE event, which could lead to a consistency of one. We find that 85% of the groups contain events from at least two vantage points, confirming that in most cases multiple vantage points qualify events as TE.

4.3 Prefixes affected by TE and origin ASes from BGP dumps

Prefixes affected by TE Recall that there are four TE techniques that we measure from BGP dumps: Selective advertisement, AS path manipulation (prepending and poisoning), and BGP action community. Finding the prefixes affected by AS path manipulation is directly visible from the AS path attribute from the BGP dump. To infer the usage of BGP action communities, we use state of the art [37], and our inference will improve as this technique improves.

We are left with evaluating our heuristic to capture selective advertisement.

To do so, we use the first dataset described in Section 4.1, where we announce our own prefixes alternatively to each of our two providers. We verify that our

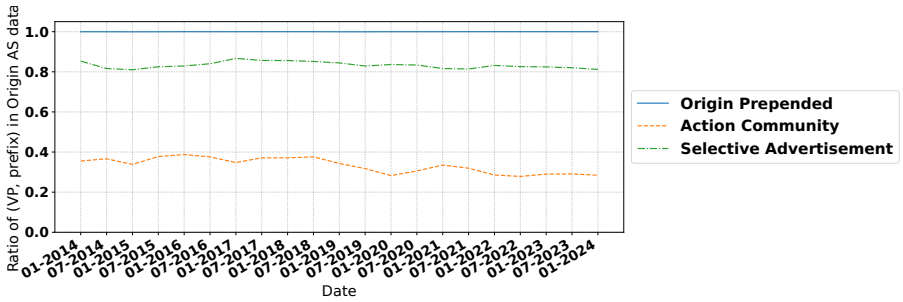

Fig. 2: Fraction of (prefix, VP) pairs with TE for which the origin AS performs the TE, over the years.

set of prefixes inferred to be part of a selective advertisement contain our three prefixes, and find that it is indeed the case. This evaluation only checks that our heuristic is able to find prefixes with selective advertisement, but does not evaluate its precision. However, it is hard to evaluate the precision of the heuristic, as the cases where the heuristic could fail are hard to reproduce. Indeed, selective advertisements can be used for blackholing or scrubbing purposes for instance. We are unlikely to see successful blackholing as such a route is usually not exported beyond the AS dropping the traffic [3]. Some scrubbing may be categorized as more-specific advertisements. For example one can redirect traffic to an off-path scrubber by advertising a more specific prefix from the scrubber. This is the exact same purpose as TE where one tries to manipulate the path followed by the traffic. We can find signs of such scrubbing by investigating prefixes with multiple origin AS or prefixes with ROAs allowing multiple origin AS.

Origin ASes performing TE For all the TE techniques, to evaluate whether we are capable of identifying that the origin AS is responsible for the TE, we compute the fraction of the (VP, prefix) pairs where the origin AS is responsible. This gives an idea of the coverage of our simple heuristics. Figure 2 shows these fractions for the different techniques over the years. Obviously, for the origin prepending technique (not shown on the graph), we are able to cover 100% of the (VP, prefix) pairs, while we are able to cover 33% of the (VP, prefix) pairs for action communities and 83% of the (VP, prefix) pairs for selective advertisement. The numbers are relatively stable over the years, varying from 29% to 39% for the action communities, and from 81% to 85% for the selective advertisement.

As a result, our study on which origin ASes perform TE (§5.3) only comprises the (VP, prefix) pairs for which we are able to infer that it is indeed the origin AS that performed TE, and therefore underestimate the number of origin ASes that are doing TE. However, these heuristics provide strong evidence that the origin AS is the one performing TE, and we choose to prioritize precision over coverage. Coverage for selective advertisement is acceptable, while we leave the

design of more sophisticated heuristics to find the AS responsible for the TE in case of the presence of an action community for future work.

Finally, we perform an analysis of the peering relationships between the origin AS performing selective advertisement and the next AS on the AS-Path for the subprefix and superprefix. We used the IYP tool [26] to fetch the AS relationships provided by BGPKIT [15]. For all (prefix, VP) over the last 10 years of sampled data with selective advertisement, we compared the relationship of the origin AS to the next AS on the AS path for the subprefix, *i.e.,* the prefix selectively advertised, and the one of the superprefix. We found that the origin AS used two different providers for both prefixes in 8,284,827 instances. The origin AS preferred using a provider for the subprefix while using a peer for the superprefix 12,898,985 times. The origin AS used a peering relationship for the subprefix and a provider for the superprefix 10,157,301. Finally, The origin AS used two different peers in 16,458,100 instances. Note that in 824,656 instances the next AS in the AS path was reported to be a downstream from the origin AS either for the subprefix or the superprefix, and, in 33,783,811 instances, we did not have any known peering relationship between the origin AS and the next AS in the AS path for either the subprefix or the superprefix. These results show that all the possible different cases exist, with no strong domination of a particular AS relationship between the origin AS and the ASes it uses to perform selective advertisement.

5 Impact of Traffic Engineering

Overview: We find that TE techniques, since 2014, are widely used by operators, for each category of ASes. In 2024, a vantage point sees on average 11% of prefixes with origin prepending, 6% of prefixes with intermediate prepending, 10% of prefixes with an action community, and 13% of prefixes with selective advertisement (§5.2). Moreover, we find that hypergiants are the category of ASes that perform TE techniques on the most prefixes, relative to the number of prefixes that they own (§5.3). Using our methodology to analyze the impact of TE on BGP stability, we find that in 2024, 39% of the BGP updates and 44% of the BGP convergence time of the BGP events are caused by TE. Finally, exploring these numbers, we find that prefixes belonging to hypergiants contribute the most to BGP instability, relative to their small number of ASes (§5.4).

5.1 Datasets and setup

We use our methodology to find TE prefixes from BGP dumps (§3.1) and from BGP updates (§3.2).

For the BGP dumps, we collect 21 snapshots over 10 years, from January 2014 to January 2024, one every six months, from all the full feed collector peers of RRC00, increasing from 18 in 2014 to 50 in 2024. For the BGP updates, we take the corresponding week of BGP updates starting at each of our BGP dumps. We discuss the sensitivity of collecting one week of updates in Section 5.4.

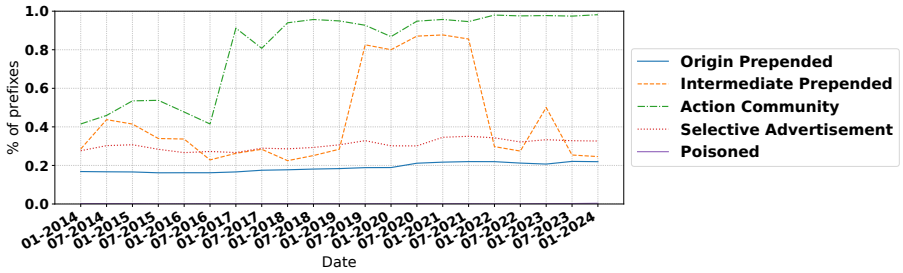

Fig. 3: Prevalence of the different TE techniques.

Then, we execute our different heuristics described in Section 3.1 on the BGP dumps to find prefixes with the different TE techniques. Finally, we run our algorithm to detect prefixes with TE events. This gives us two sets of prefixes with TE, which we call *TE prefix dumps* and *TE prefix events*.

Mapping prefixes to coarser granularities Throughout this section, we will present some results either per prefix, per AS and per AS type. To map a prefix to its AS, we simply look at the corresponding BGP dump. To map an AS to an AS type, we use the PeeringDB categorization [4], which contains 9 categories that are reported by the operators themselves. On our figures, we do not plot AS types with a very small number of ASes, such as Router Servers or Route collectors. Finally, we also remove the hypergiants from the content category to create their own category, to better emphasize some of our results about their participation in the TE. We retrieve the list of hypergiants from prior work [29]. For 2024, per category, we end up with 8,401 ASes for Cable/DSL/ISP, 3,219 ASes for Network Service Provider, 1,463 for Content, 940 from Enterprise, 138 from Hypergiant, 501 ASes for Educational/Research, 63 ASes for Government, and 294 ASes for Non-Profit. One might wonder the difference between the Cable/DSL/ISP and Network Service Provider categories. A manual look at dozens of ASes in these categories showed us that the frontier between the two categories was not clearly defined, as there are both transit providers and eyeball ASes in each category, although Network Service Provider seems to more correspond to transit while Cable/DSL/ISP to eyeball ASes. This distinction does not influence the takeaways of our results, so one can either consider it as a single or two separate categories.

5.2 Prefixes and ASes impacted by TE

We use our TE prefix dumps dataset to find which prefixes and ASes are impacted by TE, and by which TE techniques.

Prevalence of TE techniques: Figure 3 shows, over the years, the fraction of prefixes that have at least one collector peer seeing a TE technique among

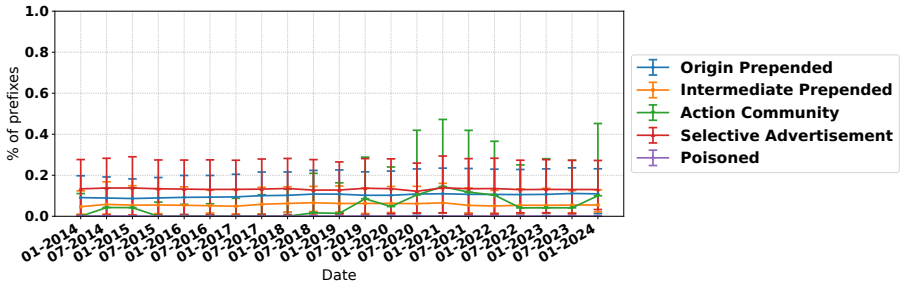

Fig. 4: Prevalence of TE techniques per VP, showing 25t, 50th, and 75th percentile.

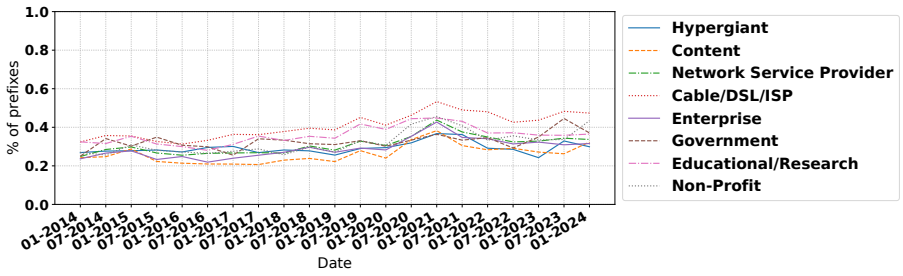

Fig. 5: Median number of prefixes affected by TE techniques per VP and per AS type. Prefixes from ASes in all AS types are affected by TE.

selective advertisement, AS path manipulation, and BGP action communities. We see that TE techniques are prevalent on the Internet, with BGP action communities being the most prevalent, with more than 99% of the prefixes affected since 2021. Path poisoning is the least prevalent, with less than 1% of the prefixes, while path prepending and selective advertisement are in between. Investigating the drop in path prepending between 2021 and 2022, we found that it was due in majority to AS 328474 and AS 174 which were prepended for 730,426 and 514,482 prefixes in 2021 and only 9 and 913 in 2022.

The 99% for action communities number is intriguing, so we complement our results per prefix with an analysis per vantage point. Figure 4 shows, over the years, the 25, 50, and 75th percentiles across the vantage points of the fraction of prefixes that are affected by a TE technique. When looking at the median, selective advertisement is the most prevalent, with more than 12% of the prefixes affected since 2014. Path poisoning is still the least prevalent, with less than 0.01% of the prefixes. However, we see that the 75th for the BGP communities is higher than for the other techniques, showing greater variability across vantage points. Indeed, we find that 2 VPs contribute to 95% of the prefixes affected by action communities.

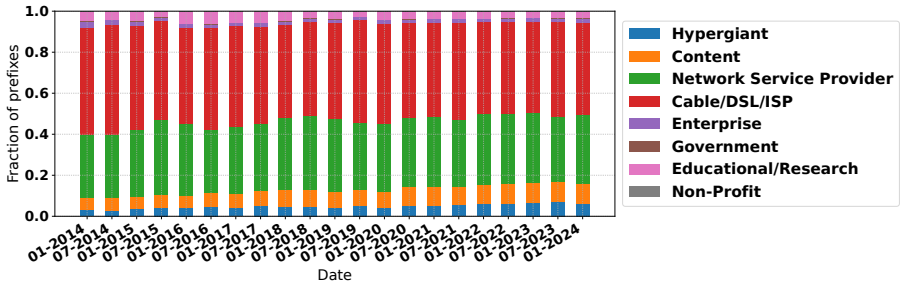

Fig. 6: Fraction of prefixes affected by TE where the origin AS is responsible for the TE, by origin AS type. Network Service Provider and Cable/DSL/ISP ASes are the ASes performing TE on the most prefixes.

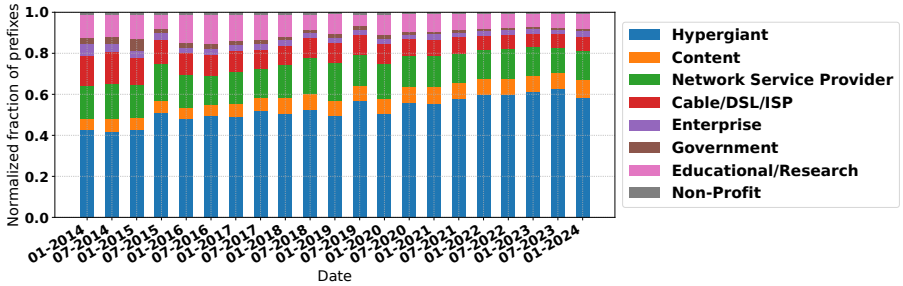

Fig. 7: Fraction of prefixes affected by TE where the origin AS is responsible for the TE, by origin AS type, normalized by the number of prefixes per origin AS type. Relatively to their small number of prefixes, Hypergiants ASes are the ASes performing TE on the most prefixes.

Prefixes and ASes affected by TE: We study the type of origin ASes that are affected by TE. We map each prefix from our two datasets of TE prefixes to their AS type, and show on Figure 5, for each AS type, the median fraction of prefixes affected by TE across the different vantage points. We observe that since 2014, all AS types are affected by TE, as the minimum fraction of prefixes over the years and across AS types is over 20%. However, there is only a small increase in the fraction of prefixes being affected across all AS types, with for instance, Hypergiants increasing from 27% to 30% between January 2014 and January 2024, and Cable/DSL/ISP increasing from 32% to 47% during the same dates.

5.3 Origin ASes performing TE

After our analysis on which prefixes are affected by TE, we now investigate which origin ASes are performing TE, using the techniques described in Section 3.1.

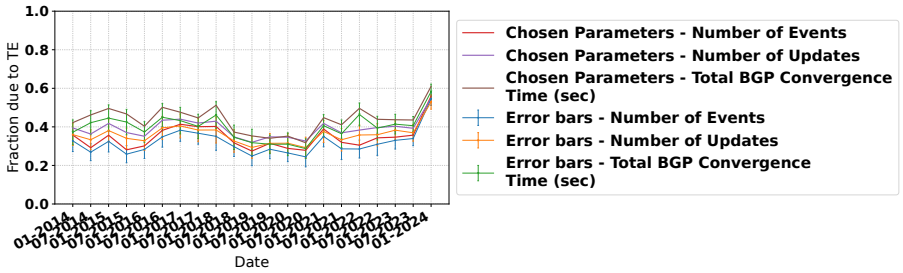

Fig. 8: Error bars of the fraction of the BGP events, BGP updates and BGP convergence time due to TE with 6 parameters set, plus the same metrics with the chosen values by our technique.

For each prefix affected by a TE technique and where the origin AS performs the TE, we compute the AS type of the origin AS. Figure 6 shows how each AS type contributes to the total number of prefixes with TE where the origin AS performs TE. We see that Network Service Provider and Cable/DSL/ISP networks are the most represented AS types, followed by Content and Hypergiants. A longitudinal analysis shows that there is a small increase of the fraction of prefixes taken by content and hypergiant, from 3% to 6% for content and from 6% to 10% for hypergiants between January 2014 to in January 2024 for content.

These results give a view of how different types of origin ASes perform TE on their prefixes. However, given that there are many more Network Service Provider and Cable/DSL/ISP ASes than in other categories, it is expected to see them practicing TE on more prefixes in absolute. To give another perspective, we normalize these results by the number of prefixes announced by each AS type. Figure 7 shows the normalized results, showing that relative to their number of prefixes, origin AS hypergiants perform more TE than other AS categories, and all over the years, varying from 42% in 2014 to 59% in 2024 of the normalized number of prefixes with TE, whereas the combination of Network Service Provider and Cable/DSL/ISP represent now 31% in 2014 and 21% in 2024 of the normalized number. For completeness, but not shown here, we mention that we computing the same plots as Figure 6 and Figure 7 per TE technique, obtaining similar results.

5.4 Impact of TE on BGP instability

We look at three metrics: (1) The number of BGP events that are due to TE (2) The number of BGP updates that are due to TE; (3) The BGP convergence time taken by TE. This analysis is performed on the TE prefix events dataset.

The number of BGP events due to TE is directly derived from our dataset. To compute the number of updates that are due to TE, we count, for each BGP event labeled as TE, the number of BGP updates between the path before and the path after the TE event. Similarly, to compute the convergence time taken

by TE, we compute the difference between the two timestamps of the two BGP updates corresponding to the starting time and the ending time of the TE event.

Instead of computing our metrics on the whole dataset of BGP updates over 10 years, we sample them, taking one week of BGP updates every six months. We evaluate how our results are sensitive to this choice. We compute the error bars of the fraction of BGP updates and BGP convergence time that are due to TE over four consecutive weeks from January 2024. We find a mean of 52% and a standard deviation of 5% for TE events, a mean of 50% and a standard deviation of 5% for BGP updates, and a mean of 54% and a standard deviation of 6% for BGP convergence time. Moreover, over the month, the fractions are 50% for BGP updates and 54% for BGP convergence time. These numbers give an idea of the sensitivity of selecting an arbitrary week and how the results can slightly vary depending on the chosen week. We perform similar analysis on weeks in the different years, finding similar results.

Then, with these results in mind, we perform a longitudinal analysis of the impact of TE on our three metrics. Figure 8 shows the fraction of the BGP events, BGP updates and BGP convergence time due to TE over the last 10 years. We can definitely say that TE significantly contributes to BGP updates and BGP convergence time. On average, over the years, 35% of the BGP events, 39% of the BGP updates, and 44% of the BGP convergence time are due to TE. We can see that the impact of TE over the years varies between 32% to 55% for BGP updates and between 32% to 61% for BGP convergence time, with the maximum attained in January 2024. However, given the sensitivity of our results to the selection of the week, we conservatively say that we cannot conclude that TE takes a more important role to BGP instability in 2024 compared to 2014, but rather say that TE just has an important role in BGP instability, looking at the absolute values of our metrics.

Sensitivity to parameters: We evaluate how the choice of the parameters alpha, the number of stable paths and the duration at which we look for those consecutive paths affect our longitudinal findings.

We ran our algorithm on the same one week sample of BGP updates over the last 10 years with 6 different parameter sets. We chose the parameters to be as conservative as possible in detecting TE. That is, we favor a high precision and choose the parameters which produce a precision of 0.99 or higher, from the sets listed in Table 2. It results the following parameter sets: (0.3, 4, 4), (0.3, 5, 4), (0.5, 4, 4), (0.5, 5, 4), (0.7, 4, 4) and (0.7, 5, 4) for alpha, the number of stable paths and the duration in hours which we look for those stable paths, respectively. Figure 8 shows error bars for the 3 metrics we study for the last 10 years. We can see that the values found using our chosen parameters (0.5, 4, 4) are similar to the average from the 6 runs with different parameters set. We found that BGP updates over the years moved from 35% to 32% when taking into account our parameters set with a 0.05 standard deviation on average. For the number of BGP updates it moved from 39% to 36% with a 0.06 standard deviation on average. Finally the average BGP convergence time due to TE moved for 44% to

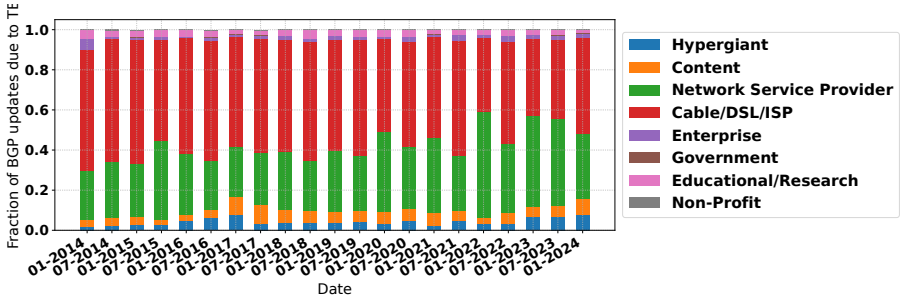

Fig. 9: Fraction of the BGP updates due to TE per origin AS type. Network Service Provider and Cable/DSL/ISP contribute the most.

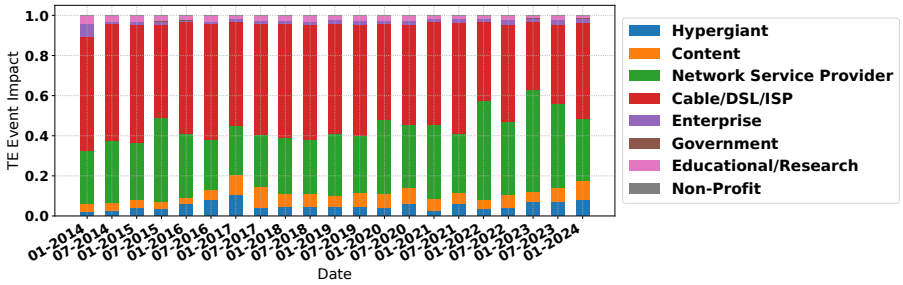

Fig. 10: Fraction of the BGP convergence time due to TE per origin AS type. Network Service Provider and Cable/DSL/ISP contribute the most.

40% with an average standard deviation of 0.06. To conclude, our results remain robust across a reasonable range of parameter values, suggesting our parameter selection is appropriate.

To which ASes the prefixes contributing the most to BGP instability belong to? After having shown global statistics about how TE affects BGP stability, we analyze to which ASes, and to which AS types the prefixes experiencing TE events belong to. To be very precise, we specifically do not phrase it as: "Which AS types contribute the most to BGP instability?" because for a prefix experiencing a TE event, we do not know whether the TE was performed by the origin AS or another AS. Therefore, our analysis shows which ASes and AS types contribute the most to BGP instability via TE because their prefixes do, but does not show which ASes actually perform the most TE operations.

For each TE event, we associate its AS and AS type as in Section 5.1. Figure 9 and Figure 10 show the part taken by the different AS types in the number of BGP updates due to TE and the part in the BGP convergence time due to TE

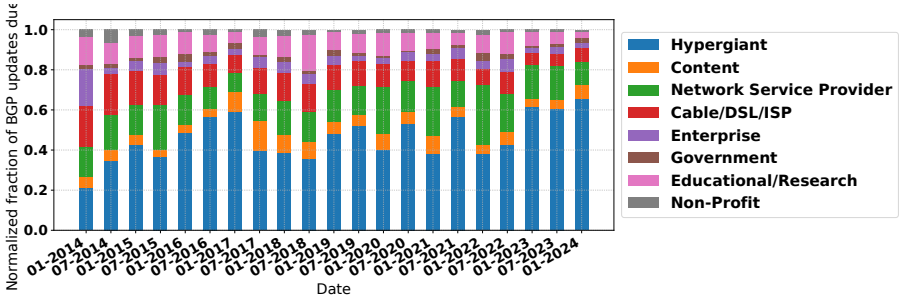

Fig. 11: Fraction of the BGP updates due to TE per origin AS type, normalized by the number of ASes per AS type. Hypergiants contribute the most.

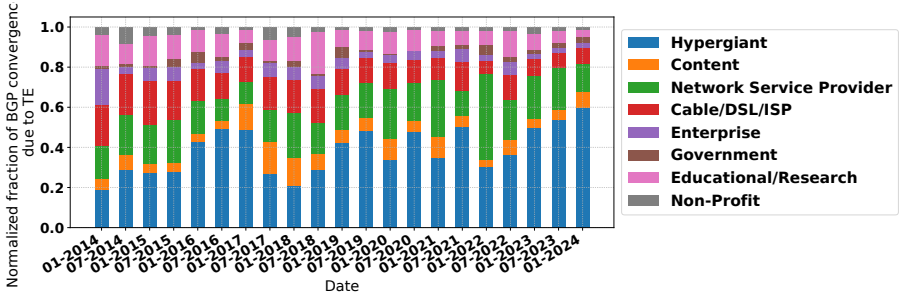

Fig. 12: Fraction of the BGP convergence time due to TE per origin AS type, normalized by the number of ASes per AS type. Hypergiants contribute the most.

over the ten last years normalized by the total number of BGP updates due to TE and the total convergence time due to TE.

We can see that prefixes from Network Service Provider and Cable/DSL/ISP dominate for all years, with, for instance, in 2024, 32% and 48% of the BGP updates due to TE, versus 8% for the next most represented category, content ASes. Logically, we obtain similar results for BGP convergence time, with 34% and 48% of the convergence time due to TE that is taken by prefixes belonging to Network Service Provider and Cable/DSL/ISP.

However, to give another perspective, Figure 11 and Figure 12 show the same dataset, but normalized per the number of AS in each category. This gives an idea of the average participation of the prefixes of an AS in an AS category to BGP instability. Similarly to what we found in Section 5.3, this normalization shows that hypergiants, relatively to their small numbers of ASes, have the biggest fraction of prefixes participating to TE instability. Indeed, they are now the first AS type, representing 66% of the normalized number of BGP updates due to TE, and 60% of the total time of BGP convergence due to TE.

Looking into more detail to which ASes belong the hypergiant prefixes, we find that the top 3 hypergiants ASes in 2024 are AS 16509 (Amazon), AS 396982

(Google), AS 13335 (Cloudflare). AS 16509 (Amazon) represents 45% of the BGP updates and 46% of the BGP convergence time of the 66% and 60% that are the contributions of hypergiants to the normalized numbers of BGP updates and BGP convergence time due to TE. AS 396982 (Google) represents 14% of the BGP updates and 20% of the BGP convergence time, while AS 13335 (Cloudflare) represents 8% of the BGP updates and 8% of the BGP convergence time.

6 Limitations

In this section we highlight potential limitations of our approach in capturing TE BGP events and in the validation of our methodology.

It is possible for an operator to change routes for all its prefixes all at once: As mentioned in Section 3.2, some operators can sometimes withdraw all the prefixes of one datacenter all at once to simulate failures. This action wouldn't be detected as TE by our algorithm.

Slow TE actions would not be captured: Our algorithm looks for stable paths within a period of a few hours. If an operator performs their path changes over a longer period of time, our algorithm will fail to capture those events.

Conservative parameter selection: The two cases described above illustrate potential limitations in our TE detection methodology. Our conservative parameter selection prioritizes precision over recall, deliberately minimizing false positives at the expense of potentially missing some TE events. Consequently, our findings should be interpreted as a lower bound on the actual prevalence of Traffic Engineering in the Internet.

Validation through research datasets: In Section 4.1, we evaluate our TE event detection algorithm using research-produced datasets. While operational BGP data would provide additional validation perspectives, these datasets enable us to assess our methodology.

7 Conclusion

In this paper we presented a principled technique to capture interdomain TE from BGP dumps and BGP updates. Our technique achieves a high precision and a high recall on our ground truth datasets. With this technique, we perform a longitudinal study of the role of TE in BGP stability, both in terms of prefixes and ASes affected, and how it impacts BGP stability. On average, over our BGP collector peers, 35% of the prefixes are affected by at least one TE technique, while we find that TE contributes to 47% of the BGP updates and 54% of the BGP convergence time.

Bibliography

[1] BGP in 2020 - BGP update churn (2007), https://blog.apnic.net/2021/01/06/bgp-in-2020-bgp-update-churn/

[2] Update about the october 4th outage (2012), https://engineering.fb.com/2021/10/04/networking-traffic/outage/

[3] Rfc 7999 (2016), https://www.rfc-editor.org/info/rfc7999

[4] Peering DB (2019), https://www.peeringdb.com/

[5] Automated Inter-AS Traffic Engineering: An open source approach and operational considerations. https://ripe85.ripe.net/presentations/10-automated-interAS-TE.pdf (2022), rIPE 85 meeting

[6] Noction Intelligent Routing Platform (Mar 2023), https://www.noction.com

[7] Cloudflare Radar (Apr 2024), https://developers.cloudflare.com/radar/

[8] Cloudflare Traffic Manager (2024), https://blog.cloudflare.com/meet-traffic-manager/

[9] IETF Draft (2024), https://www.ietf.org/archive/id/draft-ymbk-sidrops-rov-no-rr-02.html

[10] IODA (Apr 2024), https://ioda.inetintel.cc.gatech.edu

[11] NTT BGP communities (2024), https://www.gin.ntt.net/support-center/policies-procedures/routing/

[12] Afek, Y., Ben-Shalom, O., Bremler-Barr, A.: On the structure and application of bgp policy atoms. In: Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurment. pp. 209–214 (2002)

[13] Ariemma, L., Liotta, S., Candela, M., Di Battista, G.: Long-lasting sequences of bgp updates. In: International Conference on Passive and Active Network Measurement. pp. 213–229. Springer (2021)

[14] Arnold, T., Calder, M., Cunha, I., Gupta, A., Madhyastha, H.V., Schapira, M., Katz-Bassett, E.: Beating bgp is harder than we thought. In: Proceedings of the 18th ACM Workshop on Hot Topics in Networks. pp. 9–16 (2019)

[15] bigkit: Fast, extensible, on-premise global bgp monitoring (2024), https://bgpkit.com/

[16] Bischof, Z.S., Pitcher, K., Carisimo, E., Meng, A., Bezerra Nunes, R., Padmanabhan, R., Roberts, M.E., Snoeren, A.C., d: Destination unreachable: Characterizing internet outages and shutdowns. In: Proceedings of the ACM SIGCOMM 2023 Conference. pp. 608–621 (2023)

[17] Caesar, M., Rexford, J.: Bgp routing policies in isp networks. IEEE network **19**(6), 5–11 (2005)

[18] Chang, R., Lo, M.: Inbound traffic engineering for multihomed ass using as path prepending. IEEE Network **19**(2), 18–25 (2005). https://doi.org/10.1109/MNET.2005.1407694

[19] Cho, S., Fontugne, R., Cho, K., Dainotti, A., Gill, P.: Bgp hijacking classification. In: 2019 Network Traffic Measurement and Analysis Conference (TMA). pp. 25–32 (2019). https://doi.org/10.23919/TMA.2019.8784511

[20] Cisco: Configure allows-in feature in bgp (May 2024), https://www.cisco.com/c/en/us/support/docs/ip/border-gateway-protocol-bgp/112236-allowas-in-bgp-config-example.html, [Online; Updated 24-May-2024]
[21] Cittadini, L., Vissicchio, S., Donnet, B.: On the quality of bgp route collectors for ibgp policy inference. In: 2014 IFIP Networking Conference. pp. 1–9 (2014). https://doi.org/10.1109/IFIPNetworking.2014.6857091
[22] David, L., Shavitt, Y.: Bgp typo: A longitudinal study and remedies (2023)
[23] De Vries, W.B., de O. Schmidt, R., Hardaker, W., Heidemann, J., de Boer, P.T., Pras, A.: Broad and load-aware anycast mapping with verfploeter. In: Proceedings of the 2017 Internet Measurement Conference. pp. 477–488 (2017)
[24] Elmokashfi, A., Kvalbein, A., Dovrolis, C.: Bgp churn evolution: A perspective from the core. IEEE/ACM Transactions on Networking **20**(2), 571–584 (2011)
[25] Feamster, N., Borkenhagen, J., Rexford, J.: Guidelines for interdomain traffic engineering. SIGCOMM Comput. Commun. Rev. **33**(5), 19–30 (oct 2003). https://doi.org/10.1145/963985.963988, https://doi.org/10.1145/963985.963988
[26] Fontugne, R., Tashiro, M., Sommese, R., Jonker, M., Bischof, Z.S., Aben, E.: The wisdom of the measurement crowd: Building the internet yellow pages a knowledge graph for the internet. In: Proceedings of the 2024 ACM on Internet Measurement Conference. pp. 183–198 (2024)
[27] Gao, L., Rexford, J.: Stable Internet Routing without Global Coordination. In: SIGMETRICS '00 (2000)
[28] Garcia-Martinez, A., Bagnulo, M.: Measuring bgp route propagation times. IEEE Communications Letters **23**(12), 2432–2436 (2019)
[29] Gigis, P., Calder, M., Manassakis, L., Nomikos, G., Kotronis, V., Dimitropoulos, X., Katz-Bassett, E., Smaragdakis, G.: Seven years in the life of hypergiants' off-nets. In: Proceedings of the 2021 ACM SIGCOMM 2021 Conference. pp. 516–533 (2021)
[30] Gray, C., Mosig, C., Bush, R., Pelsser, C., Roughan, M., Schmidt, T.C., Wahlisch, M.: Bgp beacons, network tomography, and bayesian computation to locate route flap damping. In: Proc. ACM IMC. pp. 492–505 (2020)
[31] Gutiérrez, P.A.A.: Detection of trial and error traffic engineering with bgp-4. In: 2009 Fifth International Conference on Networking and Services. pp. 131–136 (2009). https://doi.org/10.1109/ICNS.2009.99
[32] Hauweele, D., Quoitin, B., Pelsser, C., Bush, R.: What do parrots and bgp routers have in common? ACM SIGCOMM Computer Communication Review **46**(3), 1–6 (2018)
[33] Kitabatake, T., Fontugne, R., Esaki, H.: Blt: A taxonomy and classification tool for mining bgp update messages. In: IEEE INFOCOM 2018 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS). pp. 409–414 (2018). https://doi.org/10.1109/INFCOMW.2018.8406955
[34] Krenc, T.: Bgp communities supplemental materials (2023), https://publicdata.caida.org/datasets/supplement/2023-imc-bgpcomms/

[35] Krenc, T., Beverly, R., Smaragdakis, G.: Keep your communities clean: exploring the routing message impact of bgp communities. In: Proceedings of the 16th International Conference on emerging Networking EXperiments and Technologies. pp. 443–450 (2020)

[36] Krenc, T., Beverly, R., Smaragdakis, G.: As-level bgp community usage classification. In: Proceedings of the 21st ACM Internet Measurement Conference. pp. 577–592 (2021)

[37] Krenc, T., Luckie, M., Marder, A., claffy, k.: Coarse-grained inference of BGP community intent. In: Proceedings of the 2023 ACM on Internet Measurement Conference. p. 66–72. IMC '23, Association for Computing Machinery, New York, NY, USA (2023). https://doi.org/10.1145/3618257.3624838, https://doi.org/10.1145/3618257.3624838

[38] Mahajan, R., Wetherall, D., Anderson, T.: Understanding bgp misconfiguration. In: Proceedings of the 2002 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications. p. 3–16. SIGCOMM '02, Association for Computing Machinery, New York, NY, USA (2002). https://doi.org/10.1145/633025.633027, https://doi.org/10.1145/633025.633027

[39] Marcos, P., Prehn, L., Leal, L., Dainotti, A., Feldmann, A., Barcellos, M.: As-path prepending: there is no rose without a thorn. In: Proceedings of the ACM Internet Measurement Conference. pp. 506–520 (2020)

[40] Nakamura, R., Shimizu, K., Kamata, T., Pelsser, C.: A first measurement with bgp egress peer engineering. In: Passive and Active Measurement - 23th International Conference, PAM 2022 (March 2022), https://pam2022.nl/accepted/

[41] Quan, L., Heidemann, J., Pradkin, Y.: Trinocular: Understanding internet reliability through adaptive probing. ACM SIGCOMM Computer Communication Review **43**(4), 255–266 (2013)

[42] Quoitin, B., Pelsser, C., Swinnen, L., Bonaventure, O., Uhlig, S.: Interdomain traffic engineering with BGP. IEEE Communications Magazine **41**(5), 122–128 (2003). https://doi.org/10.1109/MCOM.2003.1200112

[43] Quoitin, B., Pelsser, C., Bonaventure, O., Uhlig, S.: A performance evaluation of bgp-based traffic engineering. International Journal of Network Management **15**(3), 177–191 (2005). https://doi.org/https://doi.org/10.1002/nem.559, https://onlinelibrary.wiley.com/doi/abs/10.1002/nem.559

[44] Rizvi, A.S.M., Bertholdo, L., Ceron, J., Heidemann, J.: Anycast agility: Network playbooks to fight DDoS. In: 31st USENIX Security Symposium (USENIX Security 22). pp. 4201–4218. USENIX Association, Boston, MA (Aug 2022), https://www.usenix.org/conference/usenixsecurity22/presentation/rizvi

[45] Schlinker, B., Kim, H., Cui, T., Katz-Bassett, E., Madhyastha, H.V., Cunha, I., Quinn, J., Hasan, S., Lapukhov, P., Zeng, H.: Engineering Egress with Edge Fabric: Steering Oceans of Content to the World. In: Proc. ACM SIGCOMM (2017)

[46] Schlinker, B., Arnold, T., Cunha, I., Katz-Bassett, E.: PEERING: Virtualizing BGP at the Edge for Research. In: Proc. ACM CoNEXT '19 (2019)
[47] Sermpezis, P., Kotronis, V., Gigis, P., Dimitropoulos, X., Cicalese, D., King, A., Dainotti, A.: Artemis: Neutralizing bgp hijacking within a minute. IEEE/ACM transactions on networking **26**(6), 2471–2486 (2018)
[48] Silva Jr, B.A., Mol, P., Fonseca, O., Cunha, I., Ferreira, R.A., Katz-Bassett, E.: Automatic inference of bgp location communities. Proceedings of the ACM on Measurement and Analysis of Computing Systems **6**(1), 1–23 (2022)
[49] Streibelt, F., Lichtblau, F., Beverly, R., Feldmann, A., Pelsser, C., Smaragdakis, G., Bush, R.: Bgp communities: Even more worms in the routing can. In: Proceedings of the Internet Measurement Conference 2018. pp. 279–292 (2018)
[50] Sun, P., Vanbever, L., Rexford, J.: Scalable Programmable Inbound Traffic Engineering. In: Proceedings of the 1st ACM SIGCOMM Symposium on Software Defined Networking Research (2015)
[51] Yap, K.K., Motiwala, M., Rahe, J., Padgett, S., Holliman, M., Baldus, G., Hines, M., Kim, T., Narayanan, A., Jain, A., Lin, V., Rice, C., Rogan, B., Singh, A., Tanaka, B., Verma, M., Sood, P., Tariq, M., Tierney, M., Trumic, D., Valancius, V., Ying, C., Kallahalla, M., Koley, B., Vahdat, A.: Taking the Edge off with Espresso: Scale, Reliability and Programmability for Global Internet Peering. In: Proc. ACM SIGCOMM (2017)
[52] Zhu, J., Vermeulen, K., Cunha, I., Katz-Bassett, E., Calder, M.: The best of both worlds: high availability cdn routing without compromising control. In: Proceedings of the 22nd ACM Internet Measurement Conference. pp. 655–663 (2022)

A Ethics

Our work raises no ethical concerns.

B Outbound TE

Outbound TE techniques rather rely on the local-pref attribute or changing IGP costs. The local-pref attribute is evaluated in the first rule of the decision process. Classically its role is to favor customer over peer-to-peer over provider routes[27]. An AS dedicates a range of non-overlapping values for each class of neighboring AS. It then sets a high local-pref value from the range to the preferred neighbor in a class. The local-pref is attached to the route and propagated only inside the AS. All routers in the AS then select the exits with the highest local-pref. The local-pref is not visible outside the AS.

BGP relies on the IGP cost to pick routes with the closest exit in the AS. This is the shortest IGP cost rule. A change in the shortest paths in an AS can lead to a change of routes followed by interdomain traffic. This modification can be solely internal to the AS (and eventually noticeable by a change of location community or the presence of BGP duplicate updates) or may translate in a modification of the AS-path.