Brenes, J., Garcia-Martinez, A., Bagnulo, M., Lutu, A., Pelsser, C. (2020). Power Prefixes Prioritization for Smarter BGP Reconvergence. *IEEE/ACM Transactions on Networking*, 28(3), 1074–1087.

DOI: https://doi.org/10.1109/TNET.2020.2979665

# Power Prefixes Prioritization for Smarter BGP Reconvergence

Juan Brenes, Alberto García-Martínez and Marcelo Bagnulo, Andra Lutu, Cristel Pelsser

***Abstract*—BGP reconvergence events involving a large number of prefixes may result in the loss of large amounts of traffic. Based on the observation that a very small number of prefixes carries the vast majority of traffic, we propose Power Prefixes Prioritization (PPP) to ensure the routes of these popular BGP prefixes converge first. By doing so, we significantly reduce the amount of traffic lost during reconvergence events. To achieve this, PPP obtains an ordered list of popular prefixes through traffic inspection, and configures the resulting prefix rank in the BGP routers to prioritize the processing and advertisement of BGP routes. We model the benefits of PPP over traditional BGP processing in terms of traffic loss for both generic and a Zipf traffic distribution, and we consider the impact of sampling in the process of obtaining the prefix rank. Applying the mechanism to real traffic traces obtained from WIDE, we show that PPP reduces the amount of traffic lost by an order of magnitude, even when we configure it to use conservative sampling rates. We prototype our proposal in Quagga to show the feasibility of its implementation, and we observe similar traffic loss reduction. PPP can be deployed incrementally, as it is implemented purely as a change in the router-internal BGP processing behavior.**

***Index Terms*—BGP, Routing convergence, Traffic analysis, Traffic sampling, Zipf distribution**

## I. Introduction

The Border Gateway Protocol (BGP) enables Autonomous Systems (ASes) to exchange information about prefix reachability. BGP dynamically adapts to changes in the network, and it has been observed that the BGP convergence process can take up to 10 minutes [1]. During the time it takes for BGP to reconverge after a network or policy change, the traffic to the prefixes affected may be lost due to lack of routes or forwarding loops [2]. The amount of traffic lost is not negligible [3–5] and can undermine the quality of experience of the end user, as shown for VoIP communications [1]. Internet Service Providers (ISPs) revenues highly depend on the availability of the service they provide, which are frequently contractually defined through Service Level Agreements (SLAs) with their customers. Thus, traffic lost during BGP reconvergence events may translate into significant monetary losses for the ISPs. Not surprisingly, a majority of surveyed operators declare to care about slow convegence and take actions to prevent it [5].

In this paper, we propose the **Power Prefix Prioritization (PPP)** mechanism to alleviate high packet loss during BGP convergence by enabling routers to first process the BGP routes that carry the largest amount of traffic. We base the motivation for the Power Prefixes Prioritization mechanism on three observations, as follows.

*Observation 1:* **A single BGP event may affect a large number of routes to different prefixes.** It is very common that two ASes exchange a large number of routes through a single BGP session. A fairly common setup is when an ISP sends a full BGP feed (i.e., hundreds of thousands of routes for the global routing table) to its customer. Other arrangements, such as partial BGP feeds or peering relationships, may also result in exchanging a large number of routes. Therefore, events related with these BGP sessions, such as a link failure, a new session established or policy changes to existing sessions, may affect a large number of routes. For example, the failure of the BGP session through which a router receives a full BGP feed will cause the *Withdrawal* of hundreds of thousands of routes, and trigger the route selection process in this and other routers, which may in turn result in further advertisements.

*Observation 2:* **The time it takes for BGP to restore reachability after a BGP event that affects a large number of routes is different for each of the prefixes affected.** After a BGP event involving a large number of prefixes, BGP updates the routes for all the affected prefixes. The overall reconvergence process may take tens of seconds or more and the routes for the affected prefixes are updated at different times during this process. The reason for this is that the reconvergence process involves operations and BGP message exchanges which are not simultaneous for the prefixes involved. Traffic destined towards an affected prefix is usually lost until a new valid route for the prefix becomes available.

*Observation 3:* **A small number of BGP prefixes accounts for a large fraction of the traffic, while a large number of prefixes carry little traffic each.** It has been asserted that the distribution of Internet traffic on destination prefixes follows a highly asymmetric distribution, in particular, a Zipf distribution [6–8]. This basically means that a small number of routes carry a large proportion of the traffic, while a large number of routes carry very little traffic.

Manuscript received XXXX; revised XXXX; accepted XXXX; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor K. Argyraki. Date of publication XXXX; date of current version XXXX. The work of Alberto García-Martínez was supported by 5G Exchange, H2020-ICT-2014-2, Grant agreement ID: 671636. The work of and Marcelo Bagnulo was supported in part by the 5GinFIRE project, and by the Spanish Ministry of Economy and Competitiveness under 5GCity project, TEC2016-76795-C6-3-R. The associate editor coordinating the review of this letter and approving it for publication was XXXXXXXXX (Corresponding author: Alberto García-Martínez.)

Juan Brenes, Alberto García-Martínez and Marcelo Bagnulo are with Department of Telematics Engineering, Universidad Carlos III de Madrid, 28911 Leganés, Madrid, Spain (e-mail: jbrenesbar@gmail.com, alberto@it.uc3m.es, marcelo@it.uc3m.es).

Andra Lutu is with Telefonica Research, Telefónica Investigación y Desarrollo, Plaza Ernest Lluch i Martin, 5 Barcelona, 08019 Barcelona Spain (email: andra.lutu@telefonica.com).

Cristel Pessler is with ICube, University of Strasbourg, 300 Bd Sébastien Brant 67412 Illkirch, France. (email: pelsser@unistra.fr).

Digital Object Identifier 10.1109/TNET.XXXXXX

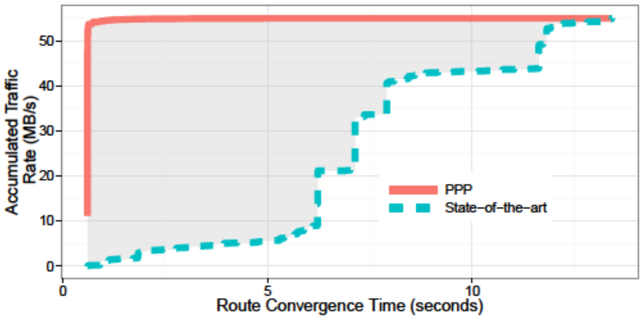


Fig. 1: Cumulative per-prefix traffic rate along the prefix convergence timeline in a BGP router

**The idea of *Power Prefix Prioritization* naturally follows after the observations above: in the case of events that affect a large number of routes, we can significantly reduce the amount of traffic loss during a failure event by ensuring that routes to traffic-intensive prefixes (which we hereinafter call *power prefixes*), converge before routes to prefixes with less traffic.** The proposed mechanism works as follows: A router samples traffic during a *measuring interval* to create a *prefix rank*, a list of prefixes ordered according to the amount traffic they carry. The BGP process uses this list for a period of time, which we call the *validity period*, to determine the route processing order for an event that simultaneously affects many routes. The asymmetric distribution of the traffic ensures that most of the traffic is recovered early in the convergence process.

In Figure 1 we show the potential benefits of PPP. The dashed line in the figure depicts the accumulated traffic rate carried by the prefixes, ordered according to their convergence time on a state-of-the-art BGP router, i.e., a router following a random order when processing BGP routes. We observe that some individual prefixes account for a large traffic share, as indicated by abrupt increases in the accumulated traffic rate, and that the times at which these *power prefixes* converge are distributed randomly all over the convergence process. The continuous line in the figure, on the other hand, shows the traffic fraction convergence timeline when the router uses PPP instead of state-of-the-art BGP. The greyed-out area between the PPP and legacy BGP convergence curves represents the difference between the accumulated traffic rate obtained in PPP and the state of the art routers. **Our experiments show that PPP provides more than one order of magnitude of traffic loss reduction in realistic scenarios.**

Although apparently simple, the idea of ordering route processing according to the amount of traffic associated to the route's prefix has never been proposed nor its feasibility proven before, as we argue in Section VII.

The remaining of the paper is structured as follows. We propose a model for the PPP mechanism and provide an approximation for the expected benefits in traffic loss reduction as a ratio of the traffic lost by PPP-enabled routers vs. the traffic lost by state-of-the-art routers (Section II). We extend the model to include the impact of the sampling process as a means to acquire the *prefix rank* (Section III). We quantify the benefits of PPP using real traffic traces from WIDE [9] (Section IV). Using this data, we obtain realistic values for different parameters of the PPP algorithm and verify the model assumptions. We then emulate PPP in typical AS topologies, to assess the effect of routing software and the interaction among multiple routers (Section V). For this, we modified the BGP Quagga routing daemon and emulated both a route reflector topology and a full-mesh topology. In both cases, PPP provided more than one order of magnitude of traffic lost reduction. In Section VI we present guidelines for the deployment of PPP in an AS. We first analyse the topologies in which a failure of a link may result in traffic loss, and then we discuss two deployment strategies. We discuss other proposals aimed to reduce traffic loss for routing events affecting many prefixes in Section VII. We argue that there are scenarios in which PPP can be seamlessly integrated with current AS configurations, while fast reroute solutions such as PIC [10, 11] cannot, notably *next-hop-self* configurations common in real deployments. We also provide quantitative comparison with two alternative proposals, TIDR [12] and Differentiated BGP Update Processing [13]. Finally, we conclude the paper in Section VIII.

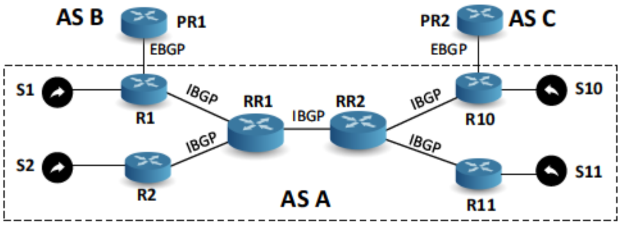


Fig. 2: AS interconnection example

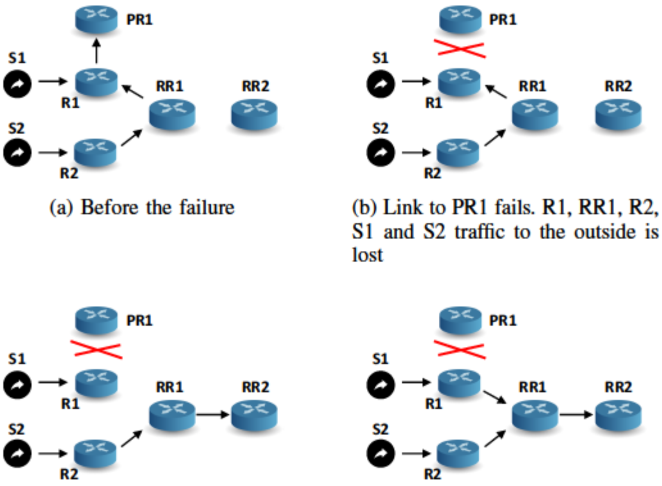
## II. Traffic Loss Model

In this section we develop an analytic model to quantify the amount of traffic lost during reconvergence in two different scenarios: (i) considering a router with the current state-of-the-art BGP implementation and (ii) considering a router using the PPP solution we propose. Using this model we provide an initial approximation of the benefits of PPP. We start analyzing the general case, in which we assume no particular traffic-to-prefix distribution. We then particularize the results for a Zipf traffic-to-prefix distribution.

### A. Traffic Loss Model: General Case

We consider an interconnection scenario where AS A has two EBGP sessions to different ASes (AS B and AS C), see Figure 2. This is just one example of topology that allows us to observe the traffic loss and we offer other examples and the conditions under which traffic loss appear in Section VI-A. AS A receives a full BGP feed of prefixes through each of the EBGP sessions established in its border routers (R1 and R10). The border routers subsequently propagate the BGP feed to route reflectors RR1 and RR2, which select the preferred routes and announces them to the other route reflector and the client routers. In our scenario, the internal routing of AS A is such that R1 prefers all the routes received from the directly connected router PR1, and RR1 prefers R1's routes due to shortest IGP distance to the next-hop router. There are four sources of traffic (S1, S2, S10, S11), each of them connected to a different router (R1, R2, R10, R11 respectively). Therefore, the traffic generated by source S2, which represents the traffic that arrives to R2 to be forwarded to the Internet, will be forwarded to RR1 and will egress through R1 to PR1. Similarly, RR2, R10 and R11 prefer the routes provided by PR2. RR1 and RR2 exchange their preferred routes, and store the routes received from the other one as alternative routes. These non-preferred routes are not propagated to its client routers R1, R2, R10 and R11. Figure 3a shows the forwarding decisions at each router located at the left part of the topology, for any external prefix.

The link between PR1 and R1 fails and reachability at routers R1 and RR1 is recovered as follows: When the failure is detected, R1 processes all prefixes sequentially, as they depended on a route received from RR1, and removes their corresponding forwarding entries, as R1 does not have alternative routes for any of them (Figure 3b). At this time, all traffic generated at S1 is discarded and also the traffic generated at S2, which is routed to R1 by RR1. In addition to the removal of the forwarding entries, R1 sends *Withdrawal* messages to inform RR1 that these destinations are no longer reachable. As the number of prefixes to process ( ) can be large, the operation to remove the routes in the routing table, sending the *Withdrawal* messages and removing the corresponding entries from the FIB takes a non-negligible time. When RR1 receives the *Withdrawal* for each prefix, it selects the alternative route through RR2 (Figure 3c). Now the traffic generated at S2, arriving at RR1, is validly forwarded through RR2; this occurs even though R2 has the old route egressing the AS through R1. Eventually, RR1 propagates the valid route through RR2 to R2 and R1. This is the time at which R1 recovers reachability through RR1-RR2-R10 (Figure 3d). In addition, RR1 propagates a *Withdrawal* to RR2, to indicate that RR2 should remove the old backup route through R1.



(a) Before the failure

(b) Link to PR1 fails. R1, RR1, R2, S1 and S2 traffic to the outside is lost

(c) RR1 receives the withdraws and reconverges on the route via RR2. Traffic for R1 and S1 is still lost

(d) All routers reconverged to the new route

Fig. 3: Forwarding state for a prefix at different stages, during a failure of the PR1-R1 link

We next analyse traffic loss in RR1, which affects the traffic generated by traffic source S2 (traffic loss in R1, i.e., affecting S1, can be analysed in a similar way). is defined as the interval since the failure occurs to the time at which the considered router RR1 installs a valid route for the first destination prefix. accounts for the time to detect the failure in R1, the time to start clearing the routing information associated with next-hop PR1 at R1, the composition and transmission of the *Withdrawal* message for the first prefix, and the route processing at RR1 to select the route for this prefix received from RR2 and install it in the forwarding plane. Besides, it also includes any other delay the messages involved could suffer, such as those caused by the MRAI timer that R1 may apply when sending the route to RR1.

Then, BGP prefixes are processed sequentially by RR1, as R1 sends *Withdrawal* messages for removing the prefixes. Prefix converges at RR1 seconds after the route for the first prefix, i.e., seconds after the link failure.

Each of the routes account for a different fraction of the traffic generated by S2 and S3. We denote as the fraction of traffic per unit of time for prefix , so that the rate for the prefix is by the total rate , the traffic generated by S2.

We can compute the traffic loss during reconvergence considering that all traffic is lost until the first *Withdrawal* is received, followed by a period in which the traffic loss depends on the order in which R1 sends the BGP updates to RR1. We call the traffic lost in this second period the *order-dependent traffic loss* and we represent it with .

$$\sum_{=1} \qquad (1)$$

Then, the total amount of traffic loss at RR1, , can be expressed as .

The PPP mechanism aims to reduce this second term , which accounts for the largest component of the traffic loss during the BGP convergence process. To achieve this, PPP routers process the prefixes according to the rank in the traffic-to-prefix distribution. This is in contrast with current BGP implementations, which walk through the prefixes in the BGP routing table following an order unrelated with the amount of traffic for each prefix.

To evaluate the benefit in using PPP, we define as the ratio between the mean *order-dependent traffic loss* with PPP, , and the mean *order-dependent traffic loss* in the BGP case, .

$$\frac{\quad}{\quad} \qquad (2)$$

We next develop a model to present expressions of and . Once the first BGP message arrives at RR1, we assume that the rest of the messages arrive sequentially, taking a fixed amount of time, , to process each prefix. In this way, the first prefix converges after , the second prefix after , and, generalizing, the announced prefix will converge after $\tau_s + n\Delta\tau$. To model current BGP implementations we consider the order in which prefixes converge to be random. We can then particularize Equation 1 as

$$RL = \sum_{i=1}^{N} t_i \; R \; X_i \; \Delta\tau \qquad (3)$$

In this and the next expressions, $i$ follows a descending order in the amount of traffic destined to each prefix, so that prefix $i$ carries more traffic per unit of time than prefix $i+1$. The random order in which R1 generates the Withdrawal messages, and thus RR1 processes the prefixes is modelled by means of $X_i$, which represents a series of random draws without replacement of integers between 1 and $N$. In other words, $X_i$ stands for the amount of time slots of duration $\Delta\tau$ that a prefix waits until router RR1 processes it.

In the PPP case, RR1 processes the $N$ prefixes according to their rank in the traffic-to-prefix distribution. The amount of traffic per unit of time for the prefix ($t_i \; R$) multiplied by the total time until the router processes it ($i \; \Delta\tau$) determines the traffic loss corresponding to the $i^{th}$ ranked prefix.

$$PL = \sum_{i=1}^{N} t_i \; R \; i \; \Delta\tau \qquad (4)$$

Note that the processing order defined by PPP is optimal: As $t_i$ decreases with $i$, any permutation of this order (e.g., with the most popular prefix converging in second position, and the second most popular prefix converging first) will result in a larger contribution to the total traffic loss. Furthermore, the higher the asymmetry of the traffic distribution, the lower the value of $PL$ and the higher the gains of using PPP compared to random order.

After removing the common term $\Delta\tau$ and computing the mean of the expression as the product of the mean of the random variable with the remaining terms, we obtain the value of $R_O$ for the traffic depending on the convergence of RR1.

$$R_O = \frac{E[\sum_{i=1}^{N} t_i \; R \; i \; \Delta\tau]}{E[\sum_{i=1}^{N} t_i \; R \; X_i \; \Delta\tau]} = \frac{2\sum_{i=1}^{N} t_i \; i}{(N+1)} \qquad (5)$$

Note tht $R_O$ is unitless, and depends exclusively on the byte-to-prefix distribution of the traffic.

We now extend this result to the data generated by source S1, directly connected to R1. In this case, R1 depends on RR1 providing an alternative route (through PR2) for every prefix affected by the failure. It is straightforward to extend the model presented for RR1 to this case, with an initial delay $\tau_{s,R1}$ for receiving the route for the first prefix, and then the sequential processing of the routes. Following the analysis in this section, we arrive to the same Expression 5. Since every source of traffic in the AS can be associated to one BGP router, we conclude that $R_O$ is an appropriate metric to evaluate the benefits of PPP in an AS regarding the way the traffic is generated in the AS.

### B. Traffic Loss Model: Zipf Distribution Case

The results of the previous subsection are applicable to any prefix-to-traffic distribution. We now model the case when the traffic is distributed on the $N$ prefixes according to a Zipf's law with scaling parameter $\alpha$. The Zipf distribution is a good approximation of the traffic-to-prefix distribution we observe in real traffic (see [6–8], and also Section IV). This law states that if we rank the $N$ prefixes according to the traffic they carry, the per-prefix traffic is inversely proportional to its rank. Then, the fraction of traffic $t_i$ for a prefix with rank $i$ is:

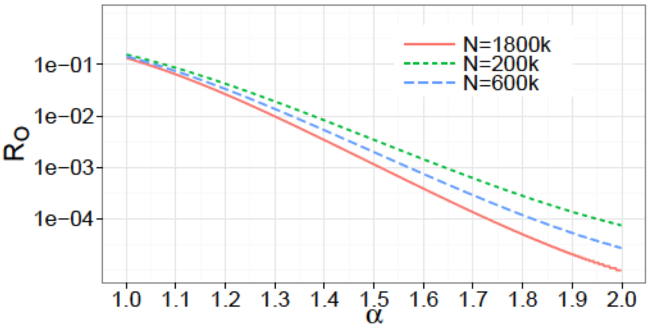


Fig. 4: $R_O$ vs. $\alpha$ for different values of $N$.

$$t_i = \frac{i^{\ \ \alpha}}{\sum_{k=1}^{N} k^{\ \ \alpha}} \qquad (6)$$

We can now replace $t_i$ (Expression 6) in Equation 5. After some simple operations, this yields the following for $R_O$:

$$R_O = \frac{2 \; \sum_{i=1}^{N} i^{1 \ \ \alpha}}{(N+1) \; \sum_{i=1}^{N} i^{\ \ \alpha}} \qquad (7)$$

In Figure 4 we plot $R_O$ as a function of the scaling parameter $\alpha$ for different number of prefixes $N$. This figure shows that PPP reduces at least one order of magnitude the traffic loss and the savings are higher when the traffic distribution is more skewed (i.e., $\alpha$ is higher). As a reference for the reader, the $\alpha$ values obtained for the dataset presented in Section IV range from 1.235 to 1.262, and the number of prefixes is between 515k and 560k.

## III. Impact of Traffic Sampling on PPP

The PPP mechanism requires a *prefix rank* to define the order in which the PPP-enabled router processes the BGP route events. The router generates the *prefix rank* after inspecting the traffic during a time interval (the *measuring interval*), under the assumption that the corresponding observed traffic-to-prefix distribution is a good predictor of the traffic in the near future (the *validity period*). Since the resources required to inspect every packet transferred in a given period of time are deemed to be too high in any practical deployment [14], traffic sampling is a requirement, and the sampling rates must be low enough to be supported by current hardware.

In this section we estimate $R_O$ when the prefix rank is obtained from sampling. We first provide an expression for $R_O$ for the case of a general packet-to-prefix and byte-to-prefix distribution, and then we particularize the results for the Zipf case. We show that PPP achieves high reduction of traffic loss with state-of-the-art packet sampling rates, confirming that the deployment of PPP is feasible.

### A. Impact of Traffic Sampling on PPP: General Case

PPP generates the *prefix rank* as follows: it first samples $s$ packets during a *measuring interval*, it then matches the destination IP address of each packet against the list of BGP prefixes in the routing table (longest prefix match), and increases the byte counter for the matched prefix with the corresponding packet length. As not every prefix may appear in the sample, we define $n_s$ as the number of prefixes identified, out of the $N$ total number of prefixes. We build the *prefix rank* with the $n_s$ prefixes ordered according to the amount traffic destined to them observed in the sample, and the rest of the prefixes that were not identified in the sample following a random order. If we define $PL_s$ as the amount of *order-dependent traffic loss* when PPP is used with this *prefix rank*, we can compute $R_O$ for the sampled case as:

$$R_O = \frac{E[PL_s]}{E[RL]} \tag{8}$$

In Equation 9, we provide an approximation of Equation 8 under any traffic distribution. The rationale for the approximation is provided in Appendix I. $p_i$ is the probability that prefix $i$ appeared in a single draw, and it depends on the *packets-to-prefix distribution* of the traffic during the *measuring interval*.

$$\begin{aligned} R_O \approx & \frac{2\ \sum_{i=1}^{N} t_i\ i\ (1 - e^{\ \ sp_i})}{(N+1)} \\ & + \frac{(N+1+\sum_{i=1}^{N}(1-e^{\ \ sp_i}))\ \sum_{i=1}^{N} t_i\ e^{\ \ sp_i}}{(N+1)} \end{aligned} \tag{9}$$

### B. Impact of Traffic Sampling on PPP: Zipf distributions

We now consider the case in which both the packets-to-prefix distribution and the bytes-to-prefix distribution follow Zipf's law, with the same scaling parameter $\alpha$.[1] Thus, for a prefix with rank $i$, the fraction of packets per time unit to prefix $i$, $C_i$, is

$$C_i = \frac{i^{\ \ \alpha}}{\sum_{k=1}^{N} k^{\ \ \alpha}} \tag{10}$$

Then, the probability of sampling a packet addressed to prefix $i$ in a single draw is $C_i$.

In Figure 5 we plot the mean $R_O$ against the sample size using Equation 9. We observe that $R_O$ improves (i.e., decreases) as the number of samples increases, up to a saturation point. This saturation point occurs when the ranking derived from the sampling process approximates to the optimal one defined by the reference distribution. The saturation values for Zipf are those resulting from Equation 7.

We highlight that low values of $R_O$ are obtained with as few as 100 samples (0.46 for $\alpha = 1.1$ and 0.20 for $\alpha = 1.5$). Values lower than 0.1 can be obtained with 1 Million samples, which requires sampling at around 12 packets per second (pps) for a sampling period of 24 hours. Commercial routers set their default sampling configuration to 1,000 packets per second (Juniper [15], Brocade [16]).

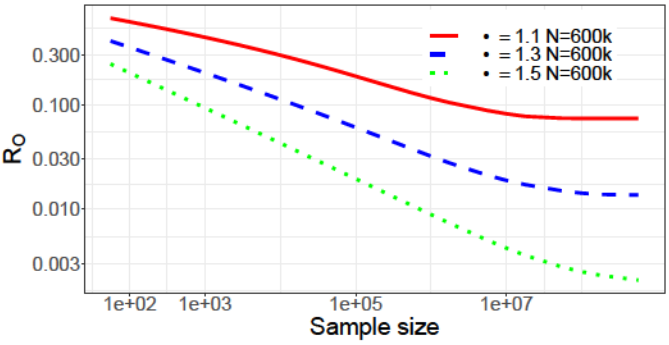


Fig. 5: $R_O$ vs sample size for different values of $\alpha$.

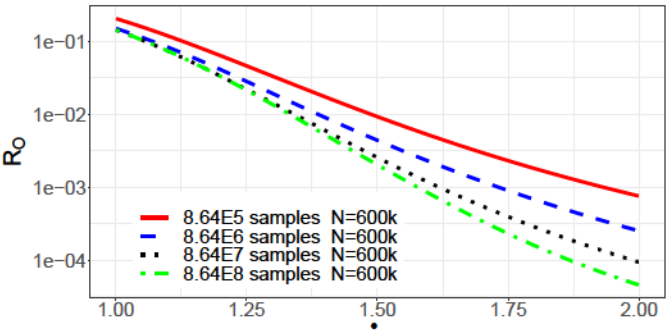


Fig. 6: $R_O$ vs $\alpha$ for different number of samples.

Figure 6 shows $R_O$ as a function of $\alpha$ (i.e., PPP benefits for different traffic distributions). The curves in the figure represent different sample sizes, selected to match the samples obtained of sampling 24 hours at 10, 100, 1000, 10 000 packets per second. In all the cases we used fixed number of prefixes $N$ of 600k, which is approximately the number of prefixes we obtained from real BGP tables for our experiment (see Section IV). We observe small variations in $R_O$ for low $\alpha$ values, while for higher $\alpha$ values, the difference grows to almost an order of magnitude. In all cases, however, the value of $R_O$ remains very low. Therefore, we conclude that PPP can bring large benefits with low enough sampling rates, consistent with the range of acceptable rates for normal router operation.

### C. Simulation-based Validation of $R_O$ for Zipf distributions

In this section, we perform simulations to validate the approximation for $R_O$ that we presented in Equation 9. We consider a Zipf distribution, and we compare the simulated results with the results obtained from Equation 9. For each combination of $\alpha$, $N$, we obtain the *prefix rank* from the draw of $s$ random samples. With this *prefix rank* and the traffic share associated to each prefix, we compute $PL_s$. We perform 40 repetitions to obtain the mean and compute $R_O$.

We select 25 different sample sizes in the interval ranging from 100 to $10^{10}$ samples[2], for $\alpha$ values of 1.1, 1.3 and 1.5, and $N$ being 200k, 600k and 1.8M prefixes. The maximum difference we obtained between Equation 9 and the corresponding simulations was 4.0%.

[1] This is consistent with experimental data we analyze (see Section IV), in which we observe that prefixes receiving a large number of bytes have a larger mean number of bytes per packet than less popular prefixes.

[2] Note that for a sampling period of 24 hours, 10 pps sampling corresponds to 8.6 $10^5$ samples, and 10 000 pps to 8.6 $10^8$ samples.

## IV. PPP Verification using real traffic traces.

We now use traffic traces to obtain realistic estimations of the gains that PPP can bring. We also use the traces to analyze the trade-offs involved in selecting the values for the parameters of the PPP algorithm, namely, the *measuring interval*, the *sampling rate* and the *validity period*. We characterize our dataset as conforming to a Zipf distribution in both traffic-to-prefix and packet-to-prefix, and compare the results for with the models presented in Sections II and III.

### A. WIDE dataset

For our analysis, we use two data sets of real traffic captures from a 1-Gbps trans-oceanic link between WIDE to one of its transit providers and the corresponding BGP routing tables from December 2014 and December 2015, respectively [9]. Each dataset contains one 24-hour-long traffic trace (from Dec. 10th, 2014 and Dec. 2nd, 2015, respectively). The mean rate of the captured traffic for this period was 53.8 MB/s at 2014 (52.3 MB/s at 2015), with 118 Kpackets/s at 2014 (94.7 Kpackets/s at 2015). Due to privacy considerations, an anonymized version of these traces is publicly available[3]. We refer to these one-day traffic traces as the *predictor dataset*. The longest prefix match algorithm associates these traces with their corresponding BGP routing entries to generate the *prefix ranks*. Figure 10 shows the byte-to-prefix rank distribution for Dec. 10th, 2014, containing 530k prefixes (560k in 2015).

Additionally we have daily 15-minute-long traffic traces for the next 20 days, taken from 2:00 PM to 2:15 PM, ISP's local time. We use them the traffic affected by routing events, to calculate the traffic loss in a simulated failure over the *reconvergence interval*. The traces include real IP addresses, so that we can match realistically the destination IP of each packet with the BGP routing information. Thus, we are able to quantify the amount of traffic volume towards a BGP prefix.

In the experiments we perform in Section V with BGP routers, where we break a link between two ASes, the fastest convergence we obtain is 15 s. Thus we use 15 s as the *reconvergence interval*. We divide each 15 minute trace in 60 non-overlapping bins of 15 s and calculate the traffic rate for each prefix. We then calculate according to the procedure described by Equation 2, with equal processing time for each prefix. In other words, we assume each prefix converges after s, in the order defined by the *prefix rank* for the particular experiment, and with the real traffic share measured in the 15-second bin of the *reconvergence interval*.

We further show results for the 2014 dataset. The results we obtained for 2015 are consistent.

### B. Measuring Interval Analysis

We aim to select a suitable *measuring interval* in order to obtain low values of . One of the key assumptions of PPP is that recent traffic is a good predictor for the *prefix rank* by the time there is a BGP reconvergence event. We use real traffic traces to assess next the impact of the *measuring interval* duration.

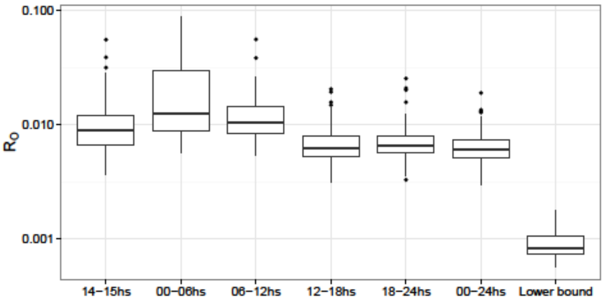


Fig. 7: for 180 15-second bins, 11th-13th Dec, 2014, 14:00 to 14:15 for different *measuring intervals*

To compute traffic loss, we divide the 15-minute traffic traces corresponding to the 3 days after the time the *prefix rank* is generated, 11th-13th Dec, 2014, into bins of 15 s, which corresponds to the *convergence period*. Then, the *prefix rank* is used to compute for the resulting 180 reconvergence bins. For now, we hypothesize that during this 3 days there is no significant variation in the traffic pattern, i.e., this interval is within the *validity period*. We validate this assumption in Section IV-D.

In order to select the best *measuring interval*, we compute the *prefix rank* with the data gathered at different periods. For now, no sampling is performed. The periods selected as *measuring intervals* are a 1-hour interval from 14:00 to 15:00 (same day period as for the traffic traces used to simulate the BGP reconvergence), four 6-hour disjoint *measuring intervals* (00-06, 06-12, 12-18, 18-24), and one 24-hour *measuring interval*. We also calculate the lower bound for PPP can achieve for each *reconvergence interval*. This lower bound is obtained using the *prefix rank* derived from the same 15-second reconvergence interval to which PPP is applied, as if we could predict exactly the traffic distribution of the prefix bin. The results for both years (Figure 7) show that all inferred *prefix ranks* behave similarly, with the 24 hour *measuring interval* resulting in a slightly better (0.006 for 2014 and 0.005 for 2015). The worst case value, i.e., the gain that can be achieved in the worst 15-second bin, is also low. It is worth to note that the 24-hour interval performs better than the 1-hour interval measured at the same period at the reconvergence intervals, which is expected to capture hourly traffic patterns that repeat daily. The lower bound for outperforms all predictors by at least 7 times (11 times for 2015). The reason for this is that the traffic pattern is more skewed at the short timescales of the reconvergence interval than at longer intervals. At the reconvergence interval timescale, we observe a lower number of active destinations, and the active destinations account for more packets per second and rate than averaged over a larger period. The communication between two parties typically involves a minimum number of packets (e.g., to initiate a TCP connection) and a minimum amount of data to exchange. Therefore, a *prefix rank* build with the exact pattern observed results in much lower traffic loss than a *prefix rank* obtained from averaging traffic for longer periods, but this *prefix rank* changes rapidly.

The two predictors fitting best are those containing the time slot in which the reconvergence event will happen, 12-18 and

[3] See traces for *samplepoint F* at http://mawi.wide.ad.jp/mawi/

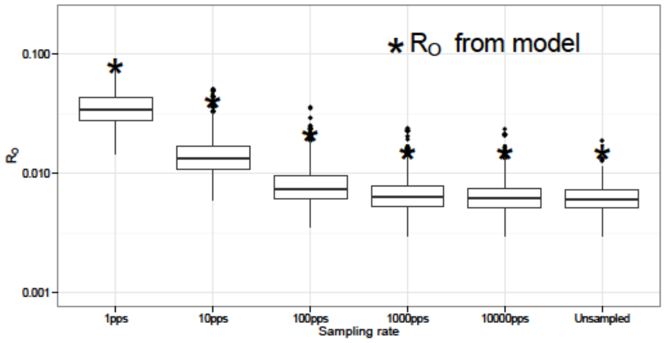


Fig. 8:  by *sampling rate*, 2014.

00-24, suggesting some mild form of hourly pattern. Since the time slot of the reconvergence event is unknown beforehand, the 00-24 predictor is expected to provide the best performance in the general case, so we use it for the rest of the analysis.

### *C. Sampling Rate Analysis*

In the previous section we omitted the impact of sampling by using all the traffic in the *measuring interval* to generate the *prefix rank*. In order to factor in the effect of traffic sampling in the *prefix rank* generation, we sample the 24-hour interval at 1, 10, 100, 1000 and 10 000 packets per second. Note that the number of packets inspected depends also of the traffic rate and the *measuring interval*.

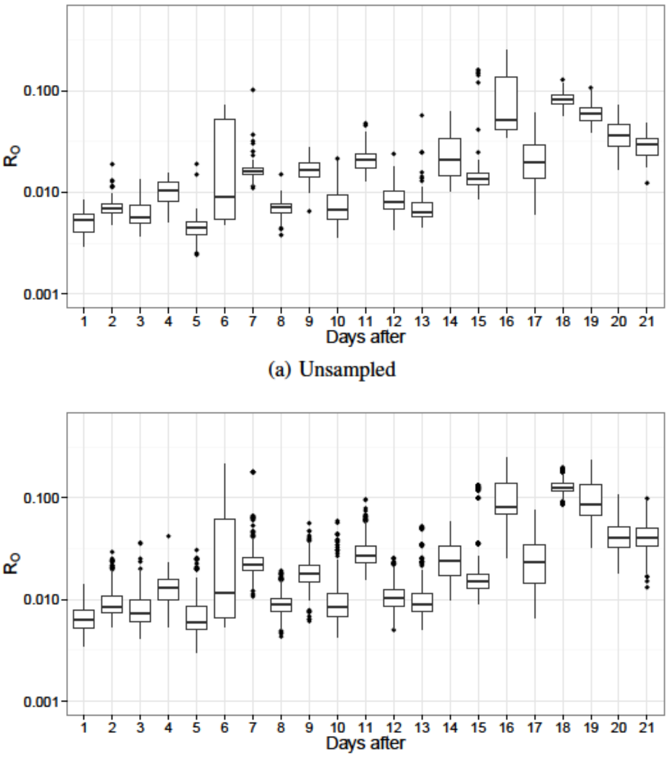
In Figure 8 we show the values obtained for  using different *sampling rates*. We observe that very low values of  can be achieved with modest *sampling rates*, since we can achieve 0.015 with a *sampling rate* of 10 pps for both years. Sampling at 10 000 pps roughly produces the same  than the unsampled case, 0.0066 for 2014 and 0.0062 for 2015.

### *D. Validity Period Analysis*

Now we investigate for how long it is reasonable to use a given *prefix rank* (i.e., to determine the *validity period* of a predictor). Figure 9 shows the evolution of  for two different 24-hour predictors of 2014, unsampled and sampled at 100 pps. This figure confirms that the predictor is valid for the first 3 days after gathering the predictor, as we assumed in the previous section. Besides, although we observe that  increases as the interval to the predictor grows, it remains below 0.15 for all the days observed. The results obtained for 2015 are fairly similar, with a maximum  of 0.2. These results suggest that PPP performs well with up to three-week *validity periods*. However, it is also true that with the low *sampling rates* needed by the PPP mechanism, the cost required to generate new *prefix ranks* is fairly low, so long *validity periods* may not be particularly attractive. We estimate that anything between one day and 3 weeks are good choices.

### *E. Model validation*

In this section we compare the results obtained in Section IV-C for  with its expected value as modeled in Sections II and III. The model for  assumes, in accordance to the existing literature [6–8], that the packet-to-prefix and byte-to-prefix are Zipf distributed, and uses the parameters of the distribution to calculate the value of  . We used the packet samples from the *predictor dataset* and the methodology described by Clauset et al. [17] to characterize these two distributions.



(a) Unsampled

(b) 100 pps

Fig. 9: Traffic lost for different days, 2014.

TABLE I: Predictor dataset characteristics

| Year | N | Packets | | | |
|---|---|---|---|---|---|
| 2014 | 515k | | 1.235 | 1.253 | 1.244 |
| 2015 | 560k | | 1.262 | 1.275 | 1.269 |

Table I resumes the information obtained. In this table,  accounts for the total number of prefixes present in the BGP table and  and  represent the scaling parameters of the packet-to-prefix and bytes-to-prefix distributions respectively. Due to the similarity among  and  , we assume that they are both equal to their mean  . In case these distributions differ in a more significant way,  should be used in the expressions at Section III-A referring to the probability  , while  should be used for the rest.

Figure 10 shows the Zipf approximation and the empirical distribution for 2014. As can be observed in this figure, only approximately 200k prefixes have traffic. Nevertheless, the contribution of the highest ranked prefixes to  is negligible as they account for a very small part of the traffic share.

Using Equation 9, we plot the theoretical value of  in Figure 8. We observe that it provides a good approximation of the values of  obtained from the data set, with real traffic exhibiting slightly better values of  than predicted by the model. One possible explanation for this is that the

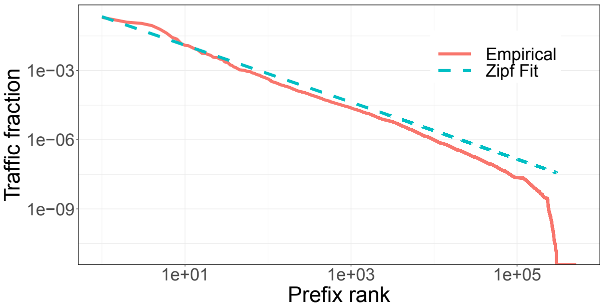


Fig. 10: Byte-to-prefix rank distribution, 2014/12/10, 24 hours.

most popular prefixes carry more traffic than the predicted by the Zipf distribution (see Figure 10). Therefore the benefits of converging those prefixes in the first stages are higher than the ones predicted by the model.

## V. Experimental Validation

In this section, we evaluate PPP with a proof-of-concept control-plane implementation. We use realistic convergence times in standard and PPP-enabled routers to compare the amount of traffic that can be saved in different configurations. For this, we modify the BGP Quagga routing daemon to process and advertise the prefixes in the order defined by the *prefix rank*, in case the BGP routing table needs to be visited completely (e.g., when a BGP session is lost).

We consider two different topologies, full-mesh and route reflector based. In both cases, we induce a failure in a link connecting an AS to one of its providers, so that the BGP reconvergence process is triggered to recover connectivity through the alternative provider. We show that the total amount of traffic lost is reduced by an order of magnitude when using PPP, compared to normal routing operation. We next detail the modification of the Quagga code, describing the configuration of the experiments. We further analyze the experiments output and the results we obtained.

### A. Modified Quagga BGP Routing Daemon

We modified the Quagga 0.99.23 BGP routing daemon `bgpd` to prioritize the processing of specific BGP prefixes when an event affecting a large number of destinations occurs. Quagga stores routing table information as a binary *trie* structure, a tree in which the binary representation of a prefix determines the position of its routing data. Routing data can be accessed in two ways, namely (i) by *prefix matching*, which is used when a BGP *Update* message is received to access the information associated with the prefix, and (ii) by *prefix iteration*, used to visit sequentially every data element of a route table. Among other cases, `bgpd` uses *prefix iteration* when it detects that a peer is no longer connected, to go over the structure holding the neighbor information, called *adj-rib-in*. In this process, it removes every route of the neighbor, selects new routes and propagates them to other neighbors.

Our modification of `bgpd`, called `bgpd-ppp`, ensures that the *prefix rank* order is followed when a *prefix iteration* is triggered. To do so, the *prefix rank* defined in a file is loaded into an array structure. Any prefix iteration starts with the first element of the array, and continues sequentially through the rest of the elements of the *prefix rank*. Once every prioritized prefix has been processed, it jumps to the prefix trie structure in which non-prioritized prefixes are stored, which is traversed in depth-first order. To support *prefix-match-based* route access, `bgpd-ppp` also stores the route information corresponding to the ranked prefixes in another trie containing both prioritized and non-prioritized prefixes trie. The full trie is used to perform prefix lookup in logarithmic time on the number of entries, instead of in linear time.

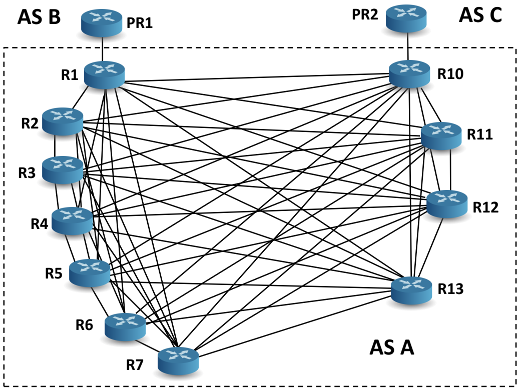


Fig. 11: Full-mesh topology

### B. Experiment execution environment and analysis

For the experiments, we deploy two AS topologies, a full-mesh topology (Fig. 11) and a route reflector topology (Fig. 14). We virtualize the scenarios using LXC, LinuX Containers, and we pin the `bgpd`/`bgpd-ppp` processes corresponding to each router to a different CPU, out of a 24 Intel Xeon E5-2620, 2.00 GHz, system. We disable the installation of BGP routes in the data plane of each node to solely focus on BGP operation[4]. Note that taking into account the installation of the routes in the data plane would result in longer convergence times, thus increasing the contribution of the *ordered-dependent traffic loss* to total traffic loss. Therefore, the results presented for total traffic loss savings are a lower bound of the values expected in equivalent real scenarios.

In each experiment we run, routers PR1 and PR2, configured as providers, propagate the same BGP route information as in the corresponding WIDE routing table snapshot. The Minimum Route Advertisement Interval (MRAI) is set to 30 s for EBGP and 5 s for IBGP, according to the default values in RFC 4271. We note that in the Quagga version used, MRAI is not applied to *Withdrawal* messages, but only to Advertisements, as stated by the BGP specification in RFC 1771 (RFC 4271, the current version, states MRAI must be applied to both types of messages). We refer the reader to the comparison of PPP

[4] If the data plane is to be tested, we would need to synthesize traffic according to the available traffic traces, along with the deployment of a topology similar to the modelled one. To complete a realistic scenario, hardware routers should be used, instead of a virtualized one in a single multiprocessor system.

with DUP (Differentiated Update Processing, see Section VII) for more detail in the effect of MRAI in traffic loss.

We compare the result of running standard Quagga `bgpd` in every router, and `bgpd-ppp` at router R1 with different *prefix ranks*. The *prefix ranks* are the ones we obtained for different sampling rates in Section IV from the 24-hour trace available for its corresponding year. Once the initial BGP convergence process completes, we disable the link PR1-R1 to model a link failure. The routers connected to PR1 and PR2 have a *BGP next-hop-self* configuration, so the IGP does not advertise through the AS that PR1 is unreachable, and route recovery depends exclusively on BGP.

In order to calculate the savings of using PPP, we compute the value of the ratio of the mean traffic loss  at a given router.  is unitless, and provides a first approximation to the gain that can be obtained by ordering route processing, regardless of the amount of traffic and the time to process BGP routes (as long as this processing time is the same for both legacy and PPP). For this, we compute the *order-dependent traffic loss* for PPP,  , using Equation 1.  , the convergence time for prefix  , is obtained from the router log traces for each experiment.  , the fraction of the total traffic that is sent to prefix  , is derived from the traffic traces of 16 15-second bins randomly selected from 180 bins belonging to the 3 days following the day of the *predictor data set*. For the non-PPP case,  is computed with the time obtained from the traces when prefixes converge without any prefix ordering. Then  is computed according to Equation 2. Since we use 4 different *prefix ranks* for each *sampling rate*, 1280 different results are obtained for each *sampling rate*. We use 5 different *sampling rates* (with 4 independent samples each) over the 24-hour *measuring interval*. We order all the prefixes with traffic to generate the *prefix rank*, and we execute 20 runs for each topology and *prefix rank*. As previously, we only show results for 2014, since 2015 shows a similar behavior.

*a) Full-mesh topology:* The internal topology between the BGP routers is a full-mesh (Figure 11), with the internal routing of AS A such that routers at the left prefer egressing through PR1, and routers at the right prefer PR2.

When PR1-R1 fails, R1 has to update the route information for every prefix for which PR1 provided a route, the full BGP feed. R1 has a backup route received from R10 (through PR2) for each destination prefix, so the router will recover connectivity for the prefix after it completes the route selection process for the considered prefix. The traffic received at R1 for each prefix is lost until the new route is selected and installed. Once R1 has a new route, it communicates to each neighbor BGP router the changes. To do so, as the new route selected was received from a router from the same AS (R10) and this route is known by every other node in a full mesh configuration, R1 just propagates a *Withdrawal* for the prefix to indicate that the route previously advertised (through the failed link) is no longer valid. When the routers that depended on R1 to exit the AS (routers R2-R7) receive this advertisement, they perform their own route selection process to install the routes egressing through PR2. Thus, we can state that all traffic for a destination is lost from the link failure until R1 installs a route through R10. From this moment, R2-R7 send traffic to the prefix to R1 (using the old route), and R1 deflects the traffic to R10. Traffic is no longer lost, although routing may be suboptimal. Eventually, R2-R7 receive the route *Withdrawal* and send their traffic directly to R10.

TABLE II:  measured for full-mesh and route reflector topologies, 2014

| Type | Sampling Rate | | | | | |
|---|---|---|---|---|---|---|
| | 1 | 10 | 100 | 1000 | 10000 | Unsampled |
| Theoretical | .0863 | .0477 | .0299 | .0239 | .0236 | .0236 |
| Experimental, R1, full-mesh | .0315 | .0117 | .0067 | .0056 | .0053 | .0051 |
| Experimental, RR1, route reflector | .0321 | .0126 | .0080 | .0074 | .0074 | .0074 |

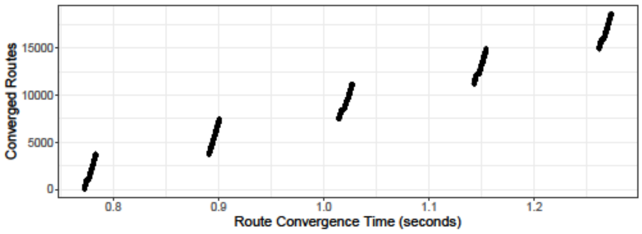
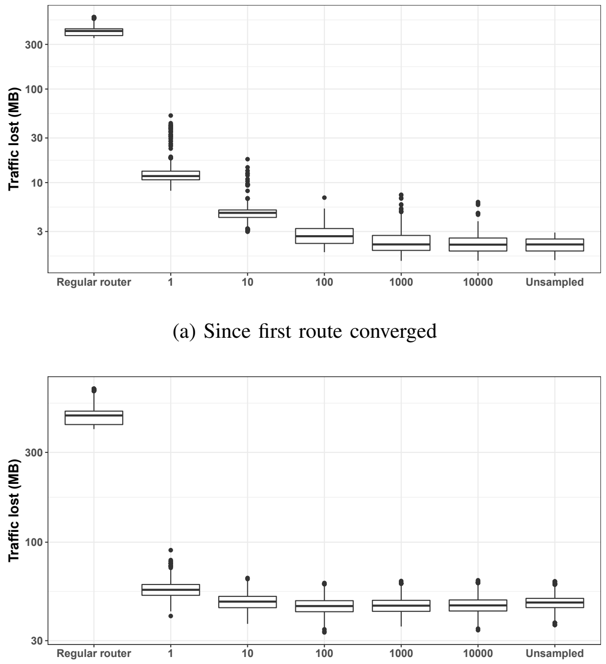


Fig. 12: Route convergence timeline for first prefixes in R1.

We now show the traffic lost, that depends on R1 convergence, as mentioned before. Note that BGP route selection at R1 takes a non-negligible time and involves a large number of prefixes with an asymmetric traffic-to-prefix distribution. Thus, the principles motivating PPP hold along with the analysis regarding to traffic loss developed in Section II. Table II shows the theoretical value of  computed for traffic loss at router R1 using the model (Equations 7 and 9). We observe that the value for  in the experiments is lower (i.e., better) than the theoretical  value. This is due to the fact that the difference between the convergence of consecutive prefixes is not constant, as assumed in the approximation but occurs as a busy period with a steady increase in the number of converged prefixes followed by a gap in which the router attends to other neighbors and/or tasks, then a busy period, etc.

Figure 12 shows the logs at R1 for a particular execution. There is an initial delay of 0.77 s until the first *Withdrawal* is sent by R1, and then multiple bursts in which the prefixes are being processed and their reachability restored. Other BGP implementations may show different behavior regarding the burst processing but we expect a similar sequential processing of routes. The first burst, encompassing the first 3736 prefixes, accounts for the 97% of the traffic if the prefixes are ordered. This burst is processed in less time than predicted by the model. Nevertheless, our model produces a good approximation for the minimum that can be expected by using PPP.

We now discuss the amount of data saved at R1 by using PPP. We consider that routers R2 to R7 generate an aggregate traffic equivalent to the total traffic measured for WIDE. Figure 13a shows the traffic lost in R1 since the first *Update* is received (the *order-dependent traffic loss*) for no prefix prioritization and prefix prioritization at R1. We observe a reduction of roughly one order of magnitude, with small differences for *sampling rates* equal to or exceeding 100 pps.



(a) Since first route converged

(b) Since link failure

Fig. 13: Total traffic lost by PPP at R2, full-mesh topology, 2014.

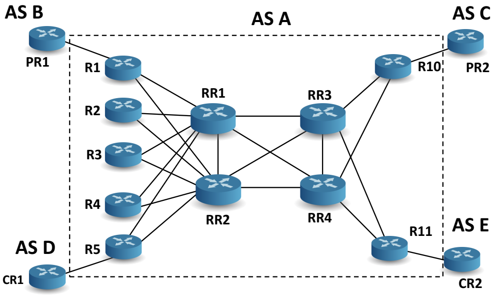
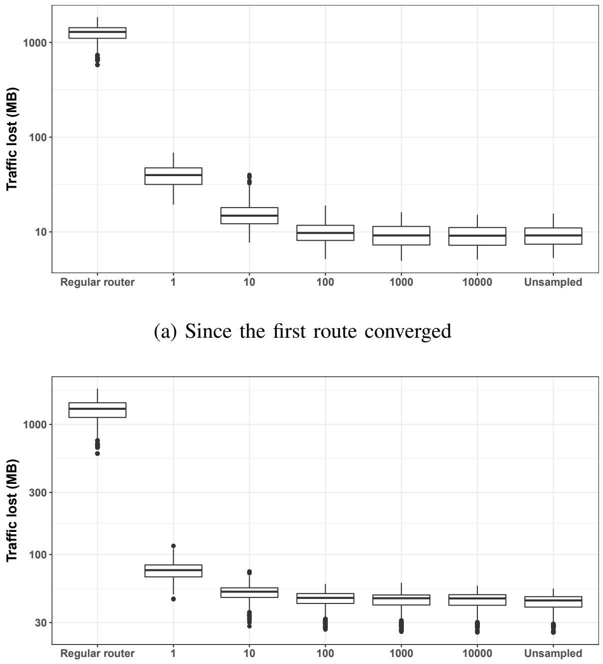


Fig. 14: Route reflector topology.

Next, we observe in Figure 13b the total amount of traffic lost, including the time to detect the failure. The amount of traffic loss with PPP ranges from 46 MB to 56 MB, while the traffic loss in a legacy router is 473 MB. Again, a reduction of an order of magnitude is achieved. The mean delay in the propagation of a route from R1 to R2-R7 is 15 ms (with a maximum observed of 110 ms), so the amount of traffic redirected through R1 (before a route through R10 is installed) is low for PPP configurations, around 1.8 MB.

*b) Route reflector topology:* In the topology of Figure 14, core routers RR1, RR2, RR3, RR4 are route reflectors (RR) connected in a full-mesh. Each access router is connected to two RRs. All links have an IGP cost of 1. We configure the AS so that all the routers on the left select PR1 as exit router for all the destinations of the Internet, based on the IGP distance to this exit point. Conversely, RR3, RR4 and all their client routers select PR2 as egress point. Since BGP ensures that only preferred routes are propagated, RR1 and RR2 receive PR2's routes, but these routes are not propagated to their clients, including to R1.



(a) Since the first route converged

(b) Since link failure

Fig. 15: Total traffic lost by PPP at RR1, route reflector topology, 2014.

The link fails and router R1 sends *Withdrawals* to both RR1 and RR2 for all the routes received from PR1. When RR1, for example, receives the *Withdrawal* for a prefix, it looks for a new route. Since the best route through R1 is no longer available, it selects the route through R1 received from RR2, which is still valid for RR1, until RR2 eventually withdraws it. Once RR1 receives the *Withdrawal* from RR2, it selects the route received from either RR3 or RR4, now being able to forward traffic to that prefix through PR2.

In terms of the ability to forward packets through the providers, once RR1 and RR2 install the route through PR2 for a given prefix, forwarding succeeds for any client router from R2 to R5, even though the BGP information at these client routers still indicates that the traffic egresses through PR1. This occurs because here the next-hop router for routers R2 to R5 is the route reflector, regardless of the egress point. The BGP convergence process continues until all routers are informed of the path in use, but reachability has already been recovered for R2 to R5. However, it is a different story for R1. Forwarding to an Internet destination at R1 is not possible until the first route for that prefix egressing through PR2 is received from either RR1 or RR2. As a conclusion, for the topology considered, the key routers to study are RR1, RR2 (which behave in a similar way) and R1.

Traffic loss for RR1, 2014, is shown in Figure 15a. with R2 to R5 generating the same traffic as WIDE. We observe a reduction of the traffic loss of up to two orders of magnitude for RR1 (and thus for routers R2 to R5, which depend on the routes of RR1 and RR2). The reduction is smaller for R1, but well above one order of magnitude. Figure 15b represents the traffic lost since the link failure, with a relative reduction of the traffic lost of roughly an order of magnitude.

Regarding to the value of , we analyse RR1, since this node is less affected than R1 by batch route processing in Quagga. We observe in Table II that the values of obtained are coherent with the values obtained for the full-mesh topology and with the corresponding approximation.

## VI. Deployment Guidelines

### A. Appropriate Topologies for PPP Deployment

In this subsection we identify the topologies and routing configurations that suffer traffic loss due to reconvergence events affecting a large number of prefixes, and thus may benefit from PPP deployment.

[2, 4] identify two possible causes for transient traffic loss in safe configurations[5] during routing events. The first cause is the lack of routing entries for the destination in the forwarding table of a router in the path. The second cause is the presence of forwarding loops, since loops may result in packet discarding due to TTL expiration or to queue drop due to the increased congestion. When any of these causes occur in the current path to the destination prefix, they say that a router is in a *data-plane failure state* for this destination.

Wang et al. [2] provides a sufficient condition for a data-plane failure in a router in case a failover occurs, such as a link failure. They define a directed graph with the preferred and alternative routes to a destination. When a link or a node fails, the graph is partitioned into the cluster of routers which had a preferred route to the destination that is not affected by the failure, and a disconnected cluster, formed by routers which had their preferred routes through the failed link. Among the disconnected cluster, we find the *cluster root*, the router directly connected to the failed link They prove that belonging to a disconnected cluster with a cluster root lacking an alternative path directly connected to a connected cluster is a sufficient condition for a router to experience a data plane failure in case of a failed link.

With this condition in mind, we can conclude that the IBGP topologies are prone to data-plane failures when the border routers do not have alternative paths through at least another border router. This is because a failure in the link connecting to the egress router will result in a failure in the cluster of routers to which it provides egress connectivity.

Several common widely used configurations fail in this category: Route Reflector topologies (as in the case shown in Figure 14) provide better scalability at the expense of limited route visibility, and as a consequence, may result in data-plane failures in case a link between an external router and its provider fails. A Route Reflector may receive several routes to a destination, but will only serve one, its own preferred route, to the BGP peers to which it is connected.

However, this is not a problem exclusive for Route Reflector topologies. Full-mesh topologies may also suffer from limited route visibility. This occurs, for example, if all the routers prefer the same egress point for a given destination, e.g., as a result of a route selection based on a local preference configuration to implement a main/backup configuration. In this case, all routers have a single route to the destination and belong to the disconnected cluster in case a failure occurs in the link connecting to the preferred route, except for those receiving an external (non-preferred) route.

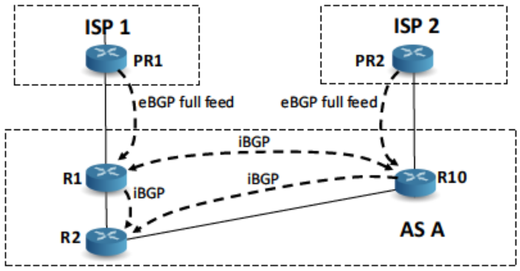


Fig. 16: AS interconnection example - BGP peering sessions

Finally, [2] note that packets may be lost during BGP convergence even if the sufficient conditions for a data-plane failure stated before do not hold. For example, in Figure 16, there is a full mesh in which routers A and C prefer their externally-received routes, so they distribute their routes to the rest. Therefore, all the routers have an alternative route to the destination. However, if the link connected to A fails, A will change its route to egress point C, but as B will temporarily prefer A (until notified of the route change), a loop can arise. Loops arise when obsolete routing information, in this case, router B using outdated router A's information, is used. This may occur whenever the logical (BGP) topology does not follow precisely the physical (data-plane) one.

### B. Deployment strategies

In this subsection, we propose two strategies for deploying PPP in a network, namely the *stand-alone* strategy and the *centralized* strategy. We also evaluate their feasibility.

The PPP *stand-alone* deployment strategy consist in the deployment of PPP in routers within an AS in a uncoordinated fashion, i.e., enabling PPP in a router does not require that other routers in the network also implement PPP. In addition, this strategy seeks for minimum configuration costs, so every PPP-enabled router autonomously generates its own *prefix rank* by inspecting the traffic traversing all of its interfaces. The PPP router samples the traffic at a given *sampling rate*, accumulating for each prefix the byte count the packets transferred during the *measuring interval*. At the end of this interval, all the prefixes are ordered according to the bytes observed, and the *prefix rank* is generated. At this point, some manually pre-configured prefixes could be inserted in the list, to reflect other priority criteria different to the bare amount of traffic. Then, the *prefix rank* is installed in the router, replacing the previous *prefix rank*, to be used for a *validity period*. The sampling process should be started again *measuring interval* time before the expiration of the *validity period*. With this setting, once configured the values of the parameters, the router does not require interaction with any external element.

Taking into account the results obtained in Section II and IV, we suggest a one-day *measuring interval*, and an equally sized *validity period*. This means that the router generates the *prefix rank* every day, with the data gathered in the last 24 h,

[5] *Safe* in the sense that they eventually converge to a stable state

which is used for the next day. This scheme is simple, results in values of close to 0.01 for the real data we observed at a sampling rate of 1000 pps, and ensures that the predictor is applied well within its *validity period* (for the next day). As observed for our data, the 24-hour period is less sensible to the time of day in which the reconvergence events occur than shorter (e.g., one hour) periods. Note that the number of samples required does not depend on the data rate, but on the byte-to-prefix and packet-to-prefix distribution. Therefore, for the number of prefixes considered in the analysis, and similar Zipf distributions, the results should be analogous. Multi-day *measuring intervals* could also be considered.

In the *stand-alone* strategy, different PPP-enabled routers may have their own *prefix ranks*, as the traffic routed through them may be different. As discussed in previous sections, the processing order is determined by the router detecting the single event (e.g., a link failure) affecting a large number of prefixes. The other routers in the AS (and beyond) process the messages as they are received. Therefore, the *prefix rank* of the router detecting the event determines the order in which messages are processed in other routers. In the worst case, a significant number of prefixes popular for a router are not popular for another router detecting a failure. To evaluate this case, traces from different locations of an AS are required, which is left for future analysis. However, we consider most prefixes to be popular in most of the routers, so we expect the impact of this case to be low.

PPP does not need to be deployed in all the routers at the same time. In order to determine the deployment strategy, we argue that PPP should be deployed first in the routers exposed to single events affecting the largest number of prefixes. Therefore, routers connected to provider ASes benefit most from PPP, since a single event (failure in the link to the provider, or failure of the provider router) affects a large number of prefixes. Once this router defines an order for processing the prefixes, according to its *prefix rank*, the rest of the routers will follow this order, extending the benefit throughout the AS, and to clients affected by the routing event.

An alternative to a *stand-alone* deployment strategy is a *centralized* deployment strategy, in which traffic from different locations is processed to generate a single *prefix rank*, which is configured in every PPP-enabled router of the AS. Traffic inspection can be performed in devices different to the routers. However, it comes at the cost of an external element and a mechanism to convey the prefix rank to the router.

Networks could define other rank criteria to ensure that most valuable prefixes converge first. A manually-defined list with the prefixes including relevant DNS servers, used for voice traffic, VPNs, etc., could be inserted in the first position of the rank. In this case, the rest of the prefixes could be ordered automatically according to the traffic-share. Besides, the prefix ranking could be derived from other automatic criteria such as the flow count per prefix.

## VII. Related Work

We start the section analysing routing mechanisms (not necessarily BGP-specific) which take advantage on the prioritization of routes, to justify that PPP is substantially different to them. We next compare PPP with proposals reducing the amount of traffic loss in the same type of routing events for which PPP is designed, i.e., BGP reconvergence affecting many of routes.

Commercial implementations of OSPF and IS-IS link state protocols enable the prioritization of certain prefixes when performing shortest path computation and route installation [18], ensuring faster convergence for certain classes of traffic more sensible to route changes such as multimedia. The number of classes is small, 3 or 4, and prefixes can not be prioritized within classes. The association between prefixes and classes is stated through explicit *access list* configuration, and can be propagated by route advertisement tags. Applied to an AS running BGP, IGP prefix prioritization could reduce the time to compute an alternative path to relevant destinations for BGP performance, such as BGP next-hops. However, IGP prefix prioritization is not able to recover connectivity to BGP destinations in case the BGP next-hop is no longer reachable, and thus it is not a replacement to PPP.

To the best of the knowledge of the authors, prefix prioritization for BGP route processing is not available in any form in the equipment of the main router vendors. Regarding to research work, only Chen et al. (TIDR [12]) have suggested a limited form of prioritization to reduce BGP churn, which may also result in lower traffic loss. TIDR, Traffic Aware Inter-domain Routing, gathers BGP prefixes into two prefix classes, *significant* and *insignificant*, according to the amount of traffic destined to the prefix. Route changes for *insignificant* prefixes are delayed for 10 minutes, so transient routes are filtered out to reduce churn. Although TIDR suggests the use of traffic statistics to prioritize BGP route processing, it does it in a fundamentally different way as PPP: TIDR induces long propagation delays, requires protocol changes, and only considers two classes in which convergence of prefixes can occur at any time within its class, while PPP uses as many classes as different prefixes.

Even if the inconvenience from the long propagation delays and the need for protocol changes where solved, the separation of traffic in two classes proposed by TIDR is an inferior solution to PPP. We compare for both PPP and TIDR, assuming equal processing time for each prefix, with the traffic-to-prefix distribution of the WIDE dataset, 2014, , . The TIDR paper suggests the *significant* prefixes should account for the top-most 90% traffic, 188 prefixes for the dataset. We assume perfect sampling, so the 188 prefixes of the *significant* class are identified accurately. *Significant* prefixes converge first, in a random order among them, then *insignificant* ones. Prefix converges according to the traffic-to-prefix rank for PPP. With this setting, TIDR provides a of 0.210, while PPP achieves 0.022, almost 10 times better.

Sun et al. [13] propose the use of *Differentiated BGP Update* (DUP) algorithms to improve routing convergence. For that purpose, they suggest a router A may accelerate the propagation to router B of the routes that are likely to be selected by B by halving in this case the value of the MRAI timer. The regular MRAI value would be set for the rest of routes sent by A, for example, for prefixes that B advertised to A, since this shows that B already has a route to the destination. Route classification can rely on additional information such as the business relationships with the neighboring AS. The authors also propose to combine DUP algorithms with a modification of the route selection process intended to select routes with highest diversity.

We apply DUP to an experiment with the route reflector topology depicted at Fig. 14, and compare it with PPP. For this, we reduce the value of the MRAI timer for the cases in which a route change affects a destination for which the neighboring router did not advertised a BGP route. We note that the Quagga version used for this experiment does only apply MRAI to Advertisement messages. Therefore, in the scenario considered, after receiving Withdrawals from R1, RR1 and RR2 propagate the newly computed routes to routers R1 to R5 with a reduced MRAI value, while keeping standard MRAI for sending to RR3 and RR4. R5 also halves the MRAI timer for the session with the customer router CR1. Since the MRAI values previously used were 30 s for EBGP sessions, and 5 s for IBGP, for this experiment they are reduced to 15 s and 2 s respectively. We execute the experiments as described in Section V, for 2014 traffic data. The total amount of the traffic loss since the link failed at router R1 when DUP is used is 85% of the traffic loss of legacy BGP. In a similar scenario, PPP reduces the amount of total traffic loss to less than 9% (sampling at 1 pps to build the predictor). The gain DUP provides is due to the reduction in the time at which the prefixes converge. However, the duration of the whole process is several times longer than the MRAI value, and traffic loss for random prefix ordering also depends on this parameter, so traffic savings are modest.

PPP can be combined with DUP to reduce further the amount of traffic loss. In the case of 1-pps sampling, the total amount of traffic loss of the combined PPP and DUP configurations is 6.4% of the traffic lost by legacy BGP (28% of improvement compared to regular PPP).

We now discuss some alternative approaches to PPP, not based on prefix prioritization, aimed to reduce the amount of traffic loss due to BGP reconvergence.

BGP convergence time, and therefore traffic loss during convergence, can be reduced by means of route architecture improvements. Hierarchical FIB (Forwarding Information Base) architectures add indirection levels to plain FIBs in order to allow many BGP route entries with the same BGP next-hop to point to the same structure in which its IP next-hop is stored [10, 19]. If a failure is detected by the IGP, and the IGP computes a new path to the BGP next-hop, only the structure holding the IP next-hop needs to be updated. This change can be very fast. This mechanism does not require any protocol modification, and can be deployed incrementally. However, the protection provided is limited to the path up to the BGP next-hop, so the links connecting with the border router of the neighboring AS with *BGP next-hop-self* configurations, for instance, are not included. Note that *BGP next-hop-self* is required in many situations, as when the address space of the AS is different from the address space used when connecting to the neighbor ASes (e.g., for BGP/MPLS IP Virtual Private Networks [20]). It is also recommended to increase stability, as it isolates the routing of the core from external events, and to reduce the information managed by the IGP [21]. PPP can be used to protect the links to external routers, and failures of the external routers itself, in *BGP next-hop-self* configurations, while it is compatible with Hierarchical FIB deployments for failures which can be solved by the IGP.

Another approach to reduce traffic loss in case of failures is to ensure that every BGP router is provisioned with more than one BGP route to every destination in both the RIB and the FIB, and to leverage the routing architecture to perform a fast switch among alternative routes.

A first step to increment path diversity in an AS is the *BGP best external* [22] configuration, which allows routers to propagate an external router when the selected route is an internal one. It does not require protocol modifications and is supported by multiple router vendors, but may overload the control plane with more routes than strictly required [19].

*Add-path* [23] extends the BGP protocol to allow the distribution of multiple routes to the same destination prefix through I-BGP. Besides requiring an upgrade of the routing software, *add-path* may be complex to configure and may result in high resource utilization [19], for example, requiring the distribution and storage of more routes for every destination.

PIC Edge [10, 11] is a routing architecture allowing the activation of an alternative route, provided by the aforementioned path diversity techniques, in case it detects the primary route is unavailable. The detection of failures depends on the IGP, as occurred for the Hierarchical FIB solution. Therefore, the same limitations apply, with the external links and neighboring AS routers left unprotected for *BGP next-hop-self* configurations. In addition, PIC Edge requires the deployment of additional techniques to extend path diversity in AS configurations with limited internal route visibility (as discussed in Section VI-A), such as *add-path*, which is not required for PPP.

SWIFT [5] uses tags to encode AS_PATH information into a Hierarchical FIB combined with a technique to timely detect link failures by inspecting bursts of BGP updates. Their prediction mechanism requires few updates to detect a link failure, and the tag scheme allows rerouting of the affected destinations, with an overall reduction of the amount of traffic loss. However, as occurred for PIC Edge, SWIFT requires the deployment of techniques to enhance path diversity in order to be useful in topologies with limited visibility.

## VIII. Conclusions

PPP, Power Prefixes Prioritization, is a novel technique to reduce traffic loss during BGP reconvergence events that affect a large number of prefixes, such as the failure of a link to a provider. PPP ensures that a router involved in such reconvergence event performs BGP prefix processing according to the estimated amount of traffic forwarded for each destination prefix, as defined by a *prefix rank* list. The benefits provided depend on the asymmetry of the traffic-to-prefix distribution, and on the ability to predict the *prefix rank* efficiently from previous measures.

We have shown the feasibility of the approach by analysing real traffic traces. The traffic observed in a real network is suitable for the mechanism, and modest *sampling rates* such as 100 or 1000 packets per second, for a *measuring interval* of 24 hours, reduces one order of magnitude the traffic loss due to the prefix ordering (which accounts for the great majority of the losses) for a *convergence intervals* as low as 15 s. Furthermore, the *prefix rank* obtained may be used for more than two weeks without great impacts on the traffic savings.

The results are consistent with the mathematical model presented, which provides an lower limit of the achievable gains as a function of the main traffic-to-prefix characterization parameters and the sampling rate. The model also shows that the gains improve as the traffic distribution is more asymmetric, and as the *sampling rate* increases. Moreover, `bgpd-ppp`, a PPP-enabled Quagga version has been deployed in two typical route reflector and full-mesh AS topologies, using traffic traces available, to show again improvements of more than one order of magnitude in a more realistic scenario, which accounts for the time to detect a failure.

PPP does not mandate any BGP protocol modification, so it can be deployed incrementally as a software update in any router of an AS as needed. We also present a standalone PPP deployment strategy which does not introduce new significant management requirements.

## ACKNOWLEDGMENTS

We want to acknowledge Kenjiro Cho, WIDE project, for providing the traces used to validate the mechanism.

## APPENDIX I

We next provide an approximation for Equation 2 when the ranking used by PPP is obtained from sampled traffic.

The traffic loss for PPP is the sum of the contributions of the traffic loss for each prefix $i$ when $s$ samples are drawn:

$$PL_s = \sum_{i=1}^{N} PL_{s,i} \tag{11}$$

$$E\left[PL_s\right] = \sum_{i=1}^{N} E\left[PL_{s,i}\right] \tag{12}$$

To compute $E\left[PL_{s,i}\right]$, we use the *law of total expectation*,

$$E\left[PL_{s,i}\right] = E\left[E\left[PL_{s,i} \mid x_{s,i}\right]\right],$$
$$x_{s,i} = \begin{cases} 1 & \text{if prefix } i \text{ appears in } s \text{ samples} \\ 0 & \text{if prefix } i \text{ does not appear in } s \text{ samples} \end{cases} \tag{13}$$

Lets denote $P(x_{s,i} = 1)$ as the probability of prefix $i$ appearing in a sample of $s$ draws. We express $E\left[PL_{s,i}\right]$ as the sum of the contributions to traffic loss of the prefixes ranked (those with $P(x_{s,i} = 1)$) and the contributions of the prefixes which were not included in the rank ($P(x_{s,i} = 0)$). Ranked prefixes converge first, according to their position in the rank; then non-ranked prefixes are processed in random order.

$$\begin{aligned} E\left[PL_{s,i}\right] = P(x_{s,i} = 1) \cdot E\left[PL_{s,i} \mid x_{s,i} = 1\right] + \\ P(x_{s,i} = 0) \cdot E\left[PL_{s,i} \mid x_{s,i} = 0\right] \end{aligned} \tag{14}$$

We approximate the contribution of the sampled prefixes by assuming that the $n_s$ different prefixes drawn are the first prefixes of the traffic distribution and appear in the same order.

$$E\left[PL_{s,i} \mid x_{s,i} = 1\right] \approx t_i \; R \cdot i \tag{15}$$

This approximation is asymptotically correct, since as $s \to \infty$, all the prefixes are sampled, and appear in their correct order.

The contribution to traffic loss in case the prefix does not appear in the sample can be computed exactly as follows:

$$\begin{aligned} E\left[PL_{s,i} \mid x_{s,i} = 0\right] = \\ \sum_{k=1}^{N} (P(n_s = k) \cdot E\left[PL_{s,i} \mid x_{s,i} = 0 \mid n_s = k\right]) = \\ \sum_{k=1}^{N} \left(P(n_s = k) \cdot t_i \; R \cdot \frac{k + N + 1}{2}\right) = \\ t_i \; R \cdot \frac{E\left[n_s\right] + N + 1}{2} \end{aligned} \tag{16}$$

Therefore, substituting Equation 15 and Equation 16 in Equation 14, and then the result in Equation 11 we obtain

$$\begin{aligned} E\left[PL_s\right] \approx \sum_{i=1}^{N} P(x_{s,i} = 1) \; t_i \; R \; i \; + \\ \sum_{i=1}^{N} (1 - P(x_{s,i} = 1)) \; t_i \; R \frac{E\left[n_s\right] + N + 1}{2} \end{aligned} \tag{17}$$

The *order-dependent traffic loss* for random ordering, which does not depend on the sampling, is

$$E\left[RL\right] = \sum_{i=1}^{N} (E\left[RL_i\right]) = \frac{N+1}{2} \sum_{i=1}^{N} t_i \; R = \frac{N+1}{2} R \tag{18}$$

Substituting Equations 17 and 18 in Equation 8, we obtain

$$\begin{aligned} R_O \approx \frac{2 \; \sum_{i=1}^{N} t_i \; i(1 - e^{\ sp_i})}{(N+1)} + \\ \frac{(N + 1 + \sum_{i=1}^{N} (1 - e^{\ sp_i})) \; \sum_{i=1}^{N} t_i \; e^{\ sp_i}}{(N+1)} \end{aligned} \tag{19}$$

We can further approximate this expression to make it depend on $p_i$, defined as the probability of a packet of prefix $i$ being selected in a single draw. The probability of a prefix appearing at least once in the samples is $P(x_{s,i} = 1) = 1 - (1 - p_i)^s$. Using the *Poisson limit theorem* for the case when $s$ is large and $p_i$ small, we obtain $(1 - p_i)^S \approx e^{\ sp_i}$. Besides, $E[n_s]$, the number of distinct prefixes identified from a draw of $s$ samples, can be computed as the sum of the probability that every element in the rank appears at least once.

$$E\left[n_s\right] = \sum_{i=1}^{N} (1 - (1 - p_i)^s) \approx \sum_{i=1}^{N} \left(1 - e^{\ sp_i}\right) \tag{20}$$

Finally, we rewrite Equation 19 as

$$\begin{aligned} R_O \approx \frac{2 \; \sum_{i=1}^{N} t_i \; i(1 - e^{\ sp_i})}{(N+1)} + \\ \frac{(N + 1 + \sum_{i=1}^{N} (1 - e^{\ sp_i})) \; \sum_{i=1}^{N} t_i \; e^{\ sp_i}}{(N+1)} \end{aligned} \tag{21}$$

## References

[1] N. Kushman, S. Kandula, and D. Katabi, “Can You Hear Me Now?!: It Must Be BGP,” *SIGCOMM Comput. Commun. Rev.*, vol. 37, no. 2, pp. 75–84, Mar. 2007.

[2] F. Wang, J. Qiu, L. Gao, and J. Wang, “On Understanding Transient Interdomain Routing Failures,” *IEEE/ACM Trans. Netw.*, vol. 17, no. 3, pp. 740–751, Jun. 2009.

[3] U. Hengartner, S. Moon, R. Mortier, and C. Diot, “Detection and Analysis of Routing Loops in Packet Traces,” in *ACM SIGCOMM IMW*, 2002.

[4] F. Wang, Z. M. Mao, J. Wang, L. Gao, and R. Bush, “A Measurement Study on the Impact of Routing Events on End-to-end Internet Path Performance,” in *ACM SIGCOMM*, 2006.

[5] T. Holterbach, S. Vissicchio, A. Dainotti, and L. Vanbever, “SWIFT: Predictive Fast Reroute,” in *ACM SIGCOMM*, 2017.

[6] J. Wallerich and A. Feldmann, “Capturing the Variability of Internet Flows Across Time,” in *INFOCOM 2006. 23-29*, 2006.

[7] N. Sarrar, S. Uhlig, A. Feldmann, R. Sherwood, and X. Huang, “Leveraging Zipf’s Law for Traffic Offloading,” *SIGCOMM Comput. Commun. Rev.*, vol. 42, no. 1, Jan. 2012.

[8] W. Shao, L. Iannone, J. Rougier, F. Devienne, and M. Viste, “Scalable BGP prefix selection for effective inter-domain traffic engineering,” in *2016 IEEE/IFIP NOMS*, 2016, pp. 315–323.

[9] C. Sony and K. Cho, “Traffic data repository at the WIDE project,” in *Proceedings of USENIX 2000 Annual Technical Conference: FREENIX Track*, 2000, pp. 263–270.

[10] Cisco, “BGP PIC Edge for IP and MPLS-VPN,” http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/iproute_bgp/configuration/xe-3s/irg-xe-3s-book/irg-bgp-mp-pic.html.

[11] Juniper, “Configuring BGP Prefix Independent Convergence for Inet,” http://www.juniper.net/documentation/en_US/junos15.1/topics/task/configuration/bgp-configuring-bgp-pic-for-inet.html, 2015.

[12] P. Chen, W. H. Cho, Z. Duan, and X. Yuan, “Traffic-aware Inter-domain Routing for Improved Internet Routing Stability,” in *Globecom*, 2008.

[13] W. Sun, Z. M. Mao, and K. G. Shin, “Differentiated BGP Update Processing for Improved Routing Convergence,” in *ICNP*, 2006.

[14] T. Zseby, “Sampling and Filtering Techniques for IP Packet Selection,” RFC 5475, Mar. 2009.

[15] Juniper, “Configuring Traffic Sampling,” http://www.juniper.net/documentation/en_US/junos16.1/topics/usage-guidelines/services-configuring-traffic-sampling.html, 2016.

[16] Brocade, “qos sflow-set-cpu-rate-limit,” http://www.brocade.com/content/html/en/command-reference-guide/fastiron-08040-commandref/, 2016.

[17] A. Clauset, C. R. Shalizi, and M. E. Newman, “Power-law distributions in empirical data,” *SIAM*, vol. 51(4), pp. 661–703, 2009.

[18] Cisco, “IS-IS Support for Priority-Driven IP Prefix RIB Installation,” http://www.cisco.com/c/en/us/td/docs/ios/12_0s/feature/guide/fslocrib.html#wp1051005, 2005.

[19] J. C. Cardona, P. Francois, B. Decraene, J. Scudder, A. Simpson, and K. Patel, “Bringing High Availability to BGP,” *Comput. Netw.*, vol. 91, no. C, pp. 788–803, Nov. 2015.

[20] Y. Rekhter and E. Rosen, “BGP/MPLS IP Virtual Private Networks (VPNs),” IETF RFC 4364, Oct. 2015.

[21] B. R. Greene and P. Smith, *Cisco ISP Essentials*. Cisco Press, 2002.

[22] P. Marques, R. Fernando, E. Chen, P. Mohapatra, and H. Gredler, “Advertisement of the best external route in BGP,” Working Draft, IETF, Internet-Draft draft-ietf-idr-best-external-05, January 2012.

[23] R. Raszuk, R. Fernando, K. Patel, D. McPherson, and K. Kumaki, “Distribution of Diverse BGP Paths,” IETF RFC 6774, Oct. 2015.

**Juan Brenes** received his degree in Telematics Engineering at U. de Montevideo, Uruguay, in 2011, and his Masters degree in Telematics at the U. Carlos III de Madrid (UC3M), Madrid, Spain in 2015. For the last four years he has been a researcher and consultant in the areas of NFV, SDN and 5G for different companies and participating in several EU funded projects. He currently works at Nextworks as Senior R&D Systems and Software Engineer, being actively involved in 5G research projects.

**Alberto García-Martínez** received a telecommunication engineering degree in 1995 and a Ph.D. in telecommunications in 1999. In 1998 he joined U. Carlos III of Madrid (UC3M), where he has been an associate professor since 2001. His main interest areas are interdomain routing, transport protocols, network security and blockchain technology. He has published more than 50 papers in technical journals, magazines, and conferences, and has also co-authored three RFCs.

**Marcelo Bagnulo** received the Electrical Engineering degree and the Ph.D. in Telecommunications in 2005, from U. Carlos III de Madrid (UC3M). He holds a tenured associate professor position at UC3M since 2008. His research interests include Internet architecture and protocols, interdomain routing and security. He has published more than 60 papers in journals and congresses and he is the author of 18 RFCs in the IETF. Dr. Bagnulo was a member of the Internet Architecture Board between 2009 and 2011.

**Andra Lutu** Andra Lutu is an Associate Researcher at Telefonica Research in Barcelona, Spain. After receiving her PhD at UC3M and IMDEA Networks Institute, she worked as a Postdoc Fellow at Simula Research Laboratory, where she was a main contributor to the H2020 MONROE project, working to build the first open European hardware infrastructure for measurements in operational mobile networks. Andra is a 2019 recipient of a H2020 MSCA Individual Fellowship grant, to work on Dynamic Interconnections for the Cellular Ecosystem (DICE).

**Cristel Pelsser** Cristel Pelsser is a professor at the University of Strasbourg since November 2015. She leads team of researchers focusing on core Internet technologies. Her aim is to facilitate network operations, avoid network disruptions and, when they occur, pinpoint the failure precisely in order to quickly fix the issue. Cristel obtained her PhD from the UcL in Belgium and spent 9 years working for ISPs.