Equilibrium Routing: from Theory to Practice

Ho Dac Duy Nguyen
Sorbonne Universités,
UPMC Univ Paris 06, UMR 7606, LIP6
Email: ho-dac-duy.nguyen@upmc.fr

Stefano Secci
Sorbonne Universités,
UPMC Univ Paris 06, UMR 7606, LIP6
Email: stefano.secci@upmc.fr

***Abstract*—Competitive routing across peering links is a notable problem in Internet routing. A few years ago, a proposal to incrementally modify the Border Gateway Protocol (BGP) decision process was done, to improve routing coordination by leveraging on the existing multi-exit discriminator BGP attribute as signaling medium among peering Internet networks. It is called Peering Equilibrium Multipath (PEMP) routing: based on a non-cooperative potential game, it can improve routing stability and efficiency while respecting unilateral routing choice, by supporting strategic multipath forwarding decisions. The contribution of this paper is twofold. First, we specify how weighted load-balancing should be done in PEMP routing and examine the benefits against even load-balancing. Then we document an implementation of PEMP routing in the Quagga open source router, better specifying some aspects. We provide a performance evaluation of the implemented PEMP routing system, showing that the computing overhead is limited.**

I. Introduction

The Internet routing system is today based on the Border Gateway Protocol (BGP) [2], which is a path-vector distributed routing protocol allowing, in the current Internet, dozens of thousands of Internet Autonomous Systems (ASes) to exchange hundreds of thousands of inter-domain paths. In its current version, BGP is such that unilateral preferences of ASes can be expressed by means of policy routing, for both inbound and outbound traffic, at the prefix and neighbor levels. After filtering routes by policy routing rules, when multiple routes are available for a same destination network prefix, the BGP decision process can avoid an arbitrary path selection either by taking the path allowing to exit your AS network at the least cost (also known as 'hot-potato'routing), or by taking the path that is preferred by the neighbor ('cold-potato'routing) on a per-neighbor basis. While the former is a purely selfish routing rule, the latter (rather altruistic) makes business sense only when the neighbor is a customer AS.

Where there is no business agreement between two interconnected ASes, and an equivalent traffic volume exchange between respective customers over both directions exists, the ASes interconnect under a so-called 'peering agreement'. In such cases, hot-potato routing can lead to quite inefficient bilateral routing solution because of the possible double application of selfish routing [3]. A few attempts in the literature try to overcome these limitations by forms of multipath routing, for example by explicit route negotiation as in [4], [5], [6], or implicit equilibrium routing as in [7], [8]. The common idea behind these works is to enlarge the set of announced BGP paths to allow improving the bilateral routing, namely in terms of routing stability. In particular, [8] proposes a non-cooperative routing equilibrium solution, called Peering Equilibrium MultiPath Routing (PEMP). It differs from other proposals in that it offers polynomial computation complexity while preserving unilateral routing preferences (leveraging on legacy external and internal gateway protocol, IGP, traffic engineering practices based on the multi-exit discriminator attribute and the IGP costs). Hence PEMP is supposed to be implementable in real systems at low computing overhead.

The contribution of this paper is twofold. First, we enhance the PEMP routing framework, addressing its load-balancing algorithm. Then, we document and evaluate its real implementation in a widely-used open source BGP router, the Quagga routing suite [16], publishing the code as open source [9]; we followed the specifications in [8], rectifying some aspects. We show that the computing overhead is indeed limited.

The paper is organized as follows. In Sect. II we introduce the necessary background. In Sect. III, we propose how to perform PEMP weighted load-balancing. We present the implemented routing system in Sect. IV. Results are presented and discussed in Sect. V. Sect. VI concludes the paper.

II. Background

The inter domain routing situation we address in this work can be considered as a particular 'competitive routing' problem. Deriving from the seminal work [10], the classical competitive routing situation is depicted in Fig. 1: a number of sources have to send traffic by a same common gateway node that is connected with parallel direct links (two links in Fig. 1) to a common destination. Each source i has to decide how much of its traffic r_i to send over which link l, i.e. ${f^i}_l$. Moreover, let each source be aware of the link cost function, i.e. $l_k({f^i}_l)$, that is convex, monotonically increasing with the overall load sent on the link: the more the load on a link, the higher the routing cost suffered by the sources transmitting on the link. In [10] it is proven that a pure-strategy routing equilibrium always exists, i.e., it is possible to decide in a stable manner how much traffic to send on which link so that each network node has no unilateral incentive to deviate from the equilibrium solution. In the specific case where there are intermediate nodes along the way to destination, the existence of equilibrium is also guaranteed but only for very specific cost functions.

Several works followed on the topic. A common contribution is to define self-enforcement protocols to decrease

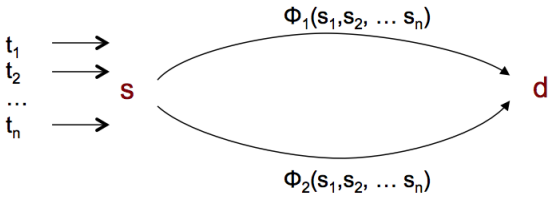

Fig. 1: Competitive routing (passive nodes)

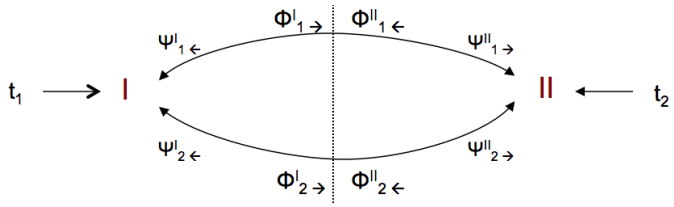

Fig. 2: Coordinated routing (active nodes)

the so-called price-of-anarchy (PoA) of the equilibrium solution, i.e., the gap between the equilibrium profile and the social optimum profile, as for example done in [12], [13]. A useful application of PoA-guided routing system design is presented in [11]: network design can be done in such a way that each network configuration is associated with the expected equilibrium routing solution, so that the best possible equilibrium routing solution guides network design choices compliant also with provider's goals. Moreover, repeated game variations of the competitive routing game are quite present in the literature. The common assumption is that a repeated interaction can more easily guarantee convergence and the efficiency of self-enforcement algorithms aiming at decreasing the PoA. Common variations consider various utility functions, which can be made sensible to interference adjustments as in [12] for wireless network situations, to destination server states as a function of the load as in [14], or to connection-level flow-control throughput and latency states as in [15].

Canonical competitive routing works are therefore particularly appropriate for applications where there is a common passive destination among multiple sources that share a common communication channel or subpath. When instead the destination is not passive but it is one among the players (see Fig. 2), as in the targeted reference peering AS scenario, the competitive routing situation is fundamentally different. When nodes in competition are both active and exchange traffic with each other, models such as those in [10]-[14] are not directly applicable. Another IP network requirement that is not easily met by legacy competitive routing approaches is that the IP link cost setting and routing decision are, in practice, two different decisions, only lightly correlated to each other, if not completely independent for some specific usages.

In Fig. 2, both nodes (I and II) are source and destination of traffic, they are autonomous decision-makers and they send traffic to each other using parallel links. The routing costs are, this time, directional costs, as traffic goes from I to II and from II to I; hence for each link and each node there are two routing costs. As such, nodes have to coordinate on the load-balancing over parallel links and the competitive routing situation can be seen as a coordinated routing problem. In the literature, approaches can be classified as negotiation-based approaches as in [4], [5], [6], and game-theoretic approaches as in [7], [8]. The former approaches target the conception of an inter-domain routing protocol supporting route proposal and acceptance/rejection signaling; in [4] a route negotiation best-reply approach is adopted, built upon bidirectional costs. In [7], instead of explicit negotiation it is proposed to exchange routing costs using in-band signaling channels; as resolution method, they propose to sum up the cost of the two players, to sort the corresponding path alternatives and then to select the shortest path. Their argument in favor of this approach, rather than a non-cooperative game equilibrium computation approach, is that the latter is NP-hard. However, in a later study [8], it is proven that preserving the unilateralism of the routing cost components as in Fig. 2 - whose value may be on different scales for different ASes (and not directly linked to the traffic load) - the resulting non-cooperative game is a special game such that an equilibrium always exists and it can be computed in a polynomial time.

The coordinated routing framework presented in [8] is called Peering Equilibrium MultiPath (PEMP) routing. It is proposed as a solution to enhance routing stability and bilateral cost across inter-AS peering links. It was specified so with marginal modifications to the current inter-domain routing protocol (BGP). More precisely, the modifications are as follows:

- *BGP signaling*: in standard BGP, the Multi-Exit Discriminator (MED) attribute can be used to suggest to an AS neighbor, connected via multiple inter-AS links, an entry point to its own AS; the MED value is typically set to the interior gateway protocol routing cost toward the destination, so that it suggests a ranking over multiple inter-AS links for a given destination IP prefix. In PEMP, it is specified to use the MED as a coordination signaling media; it is coded to transport not only the incoming routing cost, but also the outgoing routing cost.
- *BGP decision process*: when multiple routes to a same destination via a same AS exist and are considered equivalent with respect to local preference and AS hop count, the least MED rule is used to route toward the downstream AS preferred exit point. With PEMP, the least MED rule is changed so that it decides the best route or the multiple routes that correspond to the PEMP equilibria. The game components are built using the ingress/egress routing costs (four for each link, as in Fig. 2) exchanged via the MEDs.

PEMP models the inter-AS bilateral routing decision process as a 2-player non cooperative game; the two ASes act as rational players - referred to as players I and II - and the game strategy sets - X and Y - are the available peering links toward a given destination IP network. A combination of choices forms a strategy profile $(x, y) \in X \times Y$; every profile associates with a pair of unilateral payoff values that reflect the benefit of AS players associated with the corresponding routing decision. The payoff of each participant - $f(x, y)$ and $g(x, y)$, respectively - is a cost defined by the sum

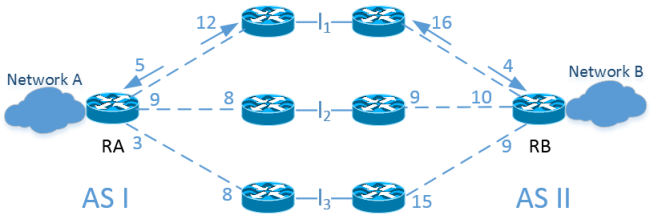

Fig. 3: Routing setting

of directional unilateral cost components. For a given AS, the egress cost component - $\phi_I(x)$ and $\phi_{II}(y)$ respectively - depends on the strategy selected by the AS itself, while the ingress cost - $\psi_I(y)$ and $\psi_{II}(x)$ - is determined by the choice of its neighbor. Hence $f(x,y) = \phi_I(x) + \psi_I(y)$ and $g(x,y) = \phi_{II}(y) + \psi_{II}(x)$.

Therefore, the resulting game $G(X,Y;f,g)$ is such that a profile indicates a link to use for each of the two players, for each of the two traffic flows from one network to the other. The two flows are considered to be equivalent, where equivalence may not strictly mean the same bit-rate, but also uneven bit-rates (as it happens in content provider to transit provider peering agreements) and even a more generic equivalence definition. This implies that at least two distinct destination IP prefixes are associated to a routing game (one for each AS), and that at most each AS associates a set of IP prefixes to the routing game. The way to segment different routing games decisions can rely on the usage of the 'BGP community' marking, which can be captured by the BGP decision process.

Under complete information sharing, both ASes can compute the same equilibrium solution. $G(X,Y;f,g)$ is a potential game, i.e., each profile (x,y) can be associated with a potential value $P(x,y)$ such that the difference in potential values between two profiles differing from an unilateral strategy move is the same independently of the other player strategy, i.e. $P(x,y) - P(x',y) = P(x,y') - P(x',y'), \quad \forall x,x' \in X, \forall y,y' \in Y$. In potential games, the minimum potential profile corresponds to a Nash equilibrium and always exists. Moreover, as proven in [8], for G all Nash equilibria always correspond to a potential minimum, which is not true for the general case. This property makes PEMP routing attractive toward realistic implementations.

It should be noted that by letting the routing decision to follow the PEMP equilibrium solution, the peering ASes reach a strategically stable routing state such that no single AS has an incentive to change its routing decision.

An example is given in Fig. 3. AS I and AS II interconnect with each other via three peering links: l_1, l_2 and l_3. As a result, router RA in AS I has three options for routing traffic from source network A to destination network B. Similarly, the same set of strategy is also available at router RB in AS II. For each intra-domain path connecting customer's network with border router, there are two internal routing costs: (a) an ingress cost represents the payoff when incoming traffic from peering AS flows on that path and (b) an egress cost indicates the payoff when forwarding packets to peer via that path. The corresponding game form is given in Table I: it summarizes all the possible outcomes of the routing game built from the above topology, it also includes the payoff and potential value of each profile. For instance, profile (l_3,l_2) has a payoff value of $(17,18)$ in which 17 is the sum of 8 and 9, that are, respectively, the routing costs at AS I when routing outgoing traffic via l_3 and receiving incoming traffic from l_2. The profiles (l_2,l_2) and (l_3,l_2) are in the Nash set. When there are multiple equilibria, if there exists a Pareto-superior one, it can be preferred as an implicit coordination rule of thumb. Otherwise, in general, load-balancing can be performed on the equilibrium profiles (as further elaborated in the next section).

TABLE I: Example game form

I\II	l_1	l_2	l_3
l_1	$(17,20)^{11}$	$(21,13)^{4}$	$(15,19)^{10}$
l_2	$(13,26)^{7}$	$\mathbf{(17,19)^{0}}$	$(11,25)^{6}$
l_3	$(13,25)^{7}$	$\mathbf{(17,18)^{0}}$	$(11,24)^{6}$

It is worth noting that, in the provided example, the routing outcome is the same as early exit (hot potato) routing, which shows that the provided framework is correctly modeling the current interconnection policies; more generally, this situation manifests when multiple profiles with the same minimum potential exists. Relying on the IGP routing cost to make routing decisions, PEMP faces the same challenge of routing instability when transient failure occurs in the intra-domain network that legacy BGP routing faces. PEMP circumvents this problem by taking into account the IGP path cost variation when deciding which profiles can eventually be considered in the routing equilibrium solution. A profile (x,y) is selected when it has potential within the minimum potential plus a threshold τ whose value is derived from the IGP path cost variation due to intra-AS link failures. Indeed, whenever a link failure happens, the costs for routing traffic across selected paths can increase. Consequently, the potential values $P(x,y)$ are recalculated, and new routing decision is made to adapt with such path cost deviation. By determining a proper threshold τ, the network operator can anticipate routing variations caused by transient failures and hence select robust equilibrium routing solutions.

III. Enhanced Load-Balancing

Leveraging on the potential sensibility and the potential threshold to fine-select routing equilibria, PEMP can alleviate the routing instability caused by hot-potato routing by preventing single equilibrium solution. When multiple equilibria exist, it is needed to develop an efficient load distribution strategy. In [8] it is proposed to perform an even load-balancing over the links corresponding to the routing equilbria. In this section, we present how to go beyond this basic rule.

For the previous example in Fig. 3, let us assume that the computed threshold value is $\tau = 4$; this implies that the profile (l_1, l_2) is also selected in the equilibrium solution, hence the related routing solution indicates load-balancing over the three peering links from AS I to AS II and single-path routing over l_2 for traffic from AS II to AS I. Performing an even load-balancing as suggested in [8], e.g., 33% on l_1, 33% on l_2, and 33% on l_3 for traffic flows from AS I to AS II, may appear in this context a rude decision as those profiles with lower potential value should attract more traffic as they are strategically more stable. It is worth recalling that a profile (x, y) is selected in the routing solution if and only if $P(x, y) \leq P_{min} + \tau$. With the purpose of minimizing the change in equilibria set before and after intra-domain failures, the value of threshold τ is computed relying on the variation of IGP path cost upon possible failures. In this way, the threshold enables to select in the routing solution the profiles that have good chances to become a pure-strategy equilibrium, i.e., which have a potential value equal or near to the minimum potential. In other words, the lower the potential value of a routing profile is, the higher the routing stability is. Distributing traffic over selected profiles equally (i.e., doing an even load balancing as specified preliminarly in [8]), does not adequately reflect this concern.

Therefore, we propose to implement an explicit PEMP load-balancing weighted as a function of the distance from the potential minimum. Let $S \in X \times Y$ be the set of selected profiles, profiles with a potential value below a threshold τ. X and Y are the set of all routing strategies available at local and peering AS respectively. The load balancing ratio for a link strategy x in X is b_x computed as (dually for b_y):

$$b_{x'} = \frac{\sum_{(x,y)\in S}^{x=x'} [1 + \tau - P(x, y)]}{\sum_{(x,y)\in S} [1 + \tau - P(x, y)]} \qquad \forall x' \in X \qquad (1)$$

The way to set the threshold initially proposed in [8] consisted in exchanging via the MED also a global directional path cost error computed as a function of link failures that could manifest at each side, taking the maximum among the minimum best path cost variations. In practice, we realized during implementation that this process would be too complex to implement, because it would add computational overhead and would mind the reliability of PEMP signaling.

We propose, instead, a more light-weight computation of the potential threshold τ for PEMP weighted load-balancing. It consists in computing a statistically relevant differential potential value corresponding to the occurrence of link failures based on known experimental failure distributions at each side. Let ΔP denote the potential difference of a strategy profile before and after an intra-domain failure. By monitoring the variation of ΔP over a number of individual link impairment scenario, a distribution of ΔP can be computed.

As an example, we apply the experimental individual link failure distribution made available in [20], which is a power-law for core links (high failure link) $n(l) \propto r(l)^{-0.73}$, in which $n(l)$ denotes the number of failures on a link $l (l = 1, ..., L)$ and $r(l)$ returns the ranking of link l with respect to its connection degree. We employ the BRITE topology generator [17] for topologies of 30 and 60 nodes, using the Barabasi and Albert BA2 model [18] and the Generalised Linear Preference (GLP) model [19]. We use a [1,20] link weight range. For every case, we repeat the failure simulation 50 times, each time with a different topology and IGP configuration. Figure 4 reports the Cumulative Distribution Function (CDF) of ΔP. It can be seen that with large topologies, the 95% ΔP is lower than 10, and for small topologies it is lower than 15, as small topologies are more subject to route instability than large topologies, as the chance that shortest path goes along a failed link is higher. It is worth noting that there is no need to have the threshold to be set exactly the same at the two borders, despite it could be a desirable routing behavior in some cases.

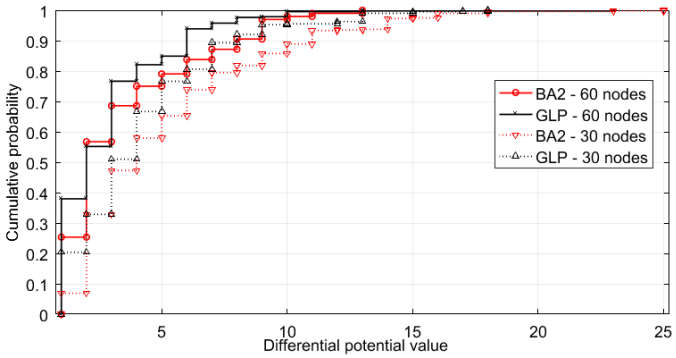

Fig. 4: CDF of ΔP in 30 and 60 nodes topologies.

With the proposed approach, determining a proper threshold is no longer a concern when considering the complexity of the PEMP routing solution for practical implementation. More important, the enhanced load-balancing technique introduced in this work offers a fair distribution over the extended set of equilibria. Forwarding a larger portion of traffic to more stable path, weighted load-balancing strategy helps to reduce the volume of traffic shifted when routing change due to transient failure. The result presented in the evaluation section justifies the effectiveness of proposed solution.

IV. Routing System Design

As already mentioned, PEMP routing reuses the MED attribute as a signaling coordination channel, and extends the BGP decision process by letting routing equilibria guiding the route selection. The forwarding decision of PEMP router is not solely destination based as in standard BGP, but it relies on both source and destination address, so modifications to the forwarding logic are also required, besides control-plane changes. Before providing more details on our PEMP implementation, we draw system-independent requirements.

A. Requirements

PEMP is an extension of the standard BGP mode that can be incrementally deployed in the current Internet. A pair of ASes willing to deploy PEMP need to just update the BGP border routers collecting the traffic from the target BGP destination cone, i.e., the set of prefixes to which apply PEMP routing (e.g.

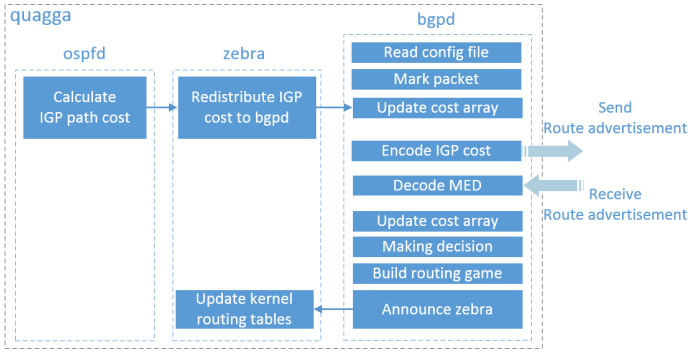

Fig. 5: System architecture of PEMP enable Quagga

marked by a BGP community). The other core BGP routers, as well as the BGP border routers at the frontier with the peering AS, do not need to be aware of PEMP routing: they just need to let MED signaling pass transparently through their filtering rules. The functional blocks to be implemented by a PEMP-enabled BGP router can be briefly summarized as follows:

- Computing directional routing cost between itself and each egress router for a given set of prefixes.
- Coding these costs into the MED attribute of corresponding route advertisements.
- Upon advertisement reception, decoding the MED and updating the routing game by considering all the possible combinations of path selections from both domains.
- Upon each setting update, determining the equilibrium routes based on the weighted load-balancing logic.
- Classifying and forwarding packets based on source and destination addresses.
- Processing inter domain routing decision and distributing load efficiently among selected paths.

B. System architecture

We enhance Quagga [16], a well-known open source routing software, more precisely its v. 0.99.23, a stable release that supports multipath routing. We choose Quagga also because differently from other common routing software like BIRD (http://bird.network.cz), it has a modular design in which each routing protocol works separately and operates as an independent process. For exchanging routing information, these processes interact and communicate with each other via a core process (ZEBRA) that plays the central role in the whole working model of the router: it summarizes routing information learned from different active protocols and frequently updates the kernel's forwarding table with new paths. The game-theoretic logic about equilibrium and load-balancing computation is externalized to an external 'routing game library (RGL)'. Our code is distributed under a GNU General public license [9].

In Fig. 5 we present the PEMP Quagga system architecture meeting the expressed requirements. To highlight the changes, we map all the new supporting functions into the original design of Quagga and hide the unaltered processes. We limit the IGP support to OSPF. Therefore the implementation of IGP path cost calculation only involves changes in the OSPFD module; it has been restructured to include ingress path cost calculation (i.e. the routing cost from each border router to the PEMP router). The other two key daemons involved are ZEBRA and BGPD. To update ZEBRA with directional path costs, we attach in the ROUTE_ADD message sent from OSPFD, the ingress cost value as well as the identification of border router. Hence we modified ZEBRA to correctly parse the received ROUTE_ADD message. With such modifications to ZEBRA and OSPFD, we meet the initial requirement for a PEMP router. Involved functions: zread_ipv4_add(), zsend_route_multipath(). In the following, we detail the major changes applied to the BGPD module to support PEMP routing.

- The routing decision is done on a per-flow basis, where a PEMP flow is defined by a pair of BGP communities: the local community of the upstream source networks, and the peer community of the downstream destination networks. The router is made able to differentiate PEMP flow traffic from normal traffic using packet marking: the classification rule is derived from a configuration file that states how to mark an incoming packet belonging to a predefined flow (to be executed by the firewall, these marking follow the FWMARK rule format). A flow-based forwarding mechanism is then needed to fulfill the requirement. Involved function: *bgp_route()*.
- Both ingress and egress cost of a routing strategy are embedded in the ROUTE_ADD message sent from ZEBRA to BGPD: the egress filtering function that automatically checks route attributes has to be customized to let the related BGP advertisement being eventually sent. The MED coding is implemented over the 32-bit value. Involved function: *bgp_redistribute_add_pemp()*.
- PEMP decoding is implemented to let the routing game data structure be built. The game structure is called every time an advertisement for a PEMP flow is detected, and is processed using the RGL methods. Involved functions: *bgp_med_decode(), bgp_pemp_game_build()*.

The above ones are control-plane enhancements. Additional forwarding plane changes are described in the following.

- In BGPD, routes determined by both the standard BGP and the PEMP decision processes are added to the same multipath route structure, where they are distinguished by the community ID attribute. BGPD then announces the multipath route to ZEBRA by a ROUTE_ADD message customized to allow attaching at each update the load-balancing weight and the community ID information. Involved functions: *bgp_best_selection(), bgp_pemp_game_build(), bgp_zebra_announce()*.
- Eventually, ZEBRA needs to update the kernel's forwarding table with routes learned from the BGP/PEMP decision process. Adaptations were needed to process the new ROUTE_ADD message format, which can include different next hops for a same destination. A separate routing table than the default table is needed as PEMP routing is source-destination based and not simply destination-based as in standard BGP. Hence we extended ZEBRA

to allow to update both types of tables, the default one and the PEMP one reserved for local community specific traffic. The target table is so identified thanks to the community ID information set as above specified. Involved functions: *zread_ipv4_add(), net_link_route_multipath()*.

One significant merit of PEMP comes from its design - rather than looking for a separated routing coordination protocol, PEMP marginally enhances the current BGP protocol by adding the necessary extensions to the signaling and decision process to allow for equilibrium routing solutions. Interoperability with legacy routers is considered as one of the crucial requirements we took into consideration when designing how to classify incoming packets, to do selective encoding IGP path cost, and to construct multiple routing tables. As we show hereafter, a PEMP-enabled router is able to work as smoothly as a legacy BGP router while performing effectively equilibrium routing for configured peering domains.

Overall, the added-in capabilities increase the total number of lines of code in Quagga by only 8%, 5% of which due to the BGPD process, the modifications in both ZEBRA and OSPFD processes being accountable for the remaining 3%). The complexity of implementing a new capability is quite interesting for developers, however it is not the right indicator for network operators that are more interested to the impact of router's performance instead.

V. Performance Evaluation

With the provided implementation, we could reproduce similar results to those presented in [8] in terms of routing cost gain, as well as routing robustness. In the following, we report novel results on the usage of the weighted load-balancing algorithm proposed in this article and on the system level performance of the PEMP implementation.

A. Weighted load-balancing vs even load-balancing

As already mentioned, in PEMP the choice on the threshold determines the routing decision stability, while the load-balancing scheme decides on the amount of traffic sent on each route. In order to evaluate the efficiency of one load-balancing scheme over the other, we examine the difference in the amount of traffic shifted during a network impairment. In this experiment, we closely monitor the change of traffic distributed at each selected path before and after a simulated failure. This measurement is applied for both weighted and even load-balancing schemes under identical conditions (same potential threshold, network topology and link failure). The failure generation follows the power-law distribution described in [20]. Network topologies are created from BRITE [17] with BA2 [18] as the modeling approach; the experiment is performed over 20 different such random topologies. At least five individual failures are generated in each topology.

In Fig. 6 we report the experiment results. Weighted load-balancing shows a better performance than even load-balancing: it has a median of 17% shifted traffic, against 26% with even load-balancing. Furthermore, its upper quartile is more than 10% smaller. Using the proposed algorithm, the load distribution ratio is derived directly from the potential value, therefore it takes into account also small variations. It is worth mentioning that weighted load-balancing shows a higher sensitivity to small variations; this is the reason why the minimum with even load-balancing is slightly lower.

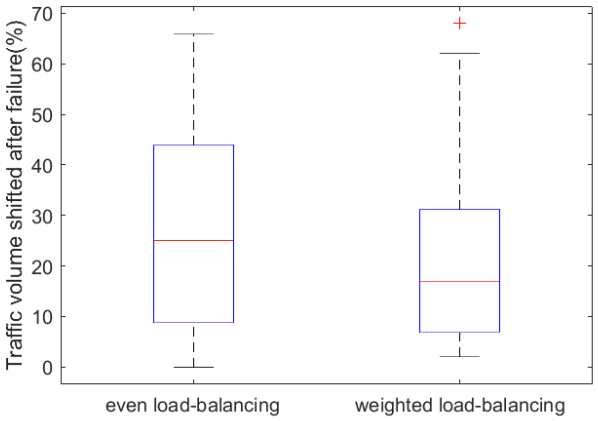

Fig. 6: Volume of traffic shifted after failure

B. System level performance

We emulate a realistic peering scenario by deploying two ASes interconnected via three peering links, using a partial mesh topology and OSPF as IGP. Each AS domain is constructed with 10 Quagga routers, among which one is configured as PEMP router and three others are selected as border routers with the neighbor AS.

We report in the following stress-test results on the PEMP routing system. We measured the performance of a router in term of processing time, i.e., the total amount of time required for processing PEMP network/link state updates and for installing new routing decision, for an increasing data-plane traffic load. The experimented routers are built in Ubuntu virtual machines with two 2.397GHz CPUs and 8GB of live memory. Two experiments are conducted to study the overheads of PEMP solution in different scenarios.

In the first experiment, we measure the processing time of router in case of OSPF path cost changes. This time typically is due to the time to recompute the IGP shortest paths and costs, to update the BGP states and to issue (possibly new) BGP routing decisions depending on the IGP costs. With PEMP, extra marginal delays are introduced for ingress cost calculation, local IGP path cost update, and game building processes. We aim to have an experimental evaluation of the total PEMP execution time overhead. It is worth noting that the current BGP implementation in Quagga waits for a periodic update process that runs every 60s to capture IGP path cost variations: we subtracted this constant time to focus on the marginal time increase. As depicted in Fig. 7, the average processing time of both PEMP and BGP are rising gradually as the data-plane traffic increases. Unsurprisingly, the standard BGP router always shows a better performance than its extension. The processing time gap is, however, quite

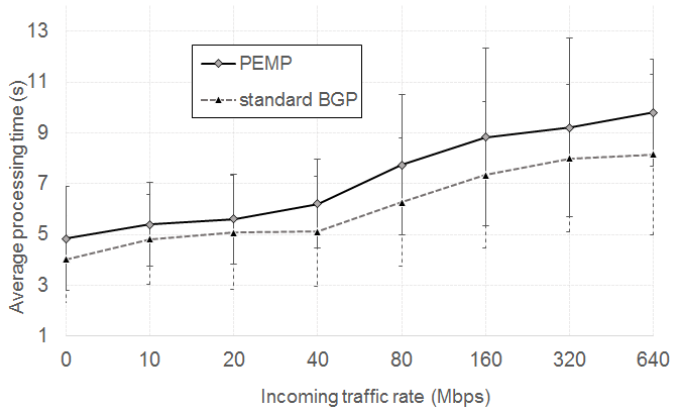

Fig. 7: Average processing time upon IGP path cost change.

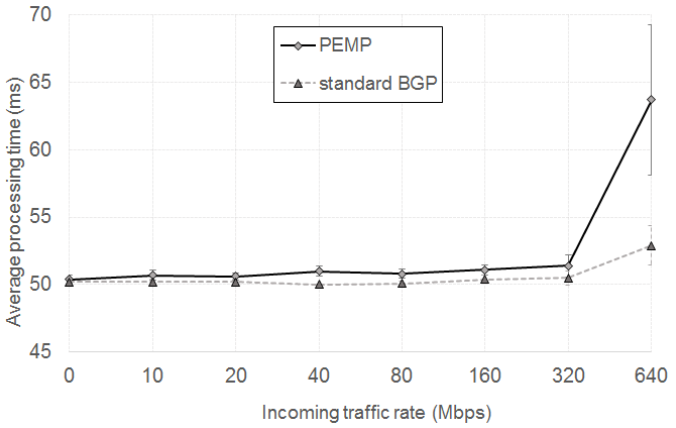

Fig. 8: Average processing time upon MED attribute change.

limited, about 15%, and regardless of incoming bitrate. As observed from the experiments, the IGP path cost update phase was the most time consuming task. With PEMP, the delay for path cost calculation is higher than with standard BGP because it needs to calculate the ingress path cost to each egress point. We believe this phase could be improved by code optimization to make this step faster.

In the second experiment, we measure the BGP router processing time in case of MED-icated route updates. Differently from the previous experiment, changing MED signaling is handled right upon reception. By default, once a MED value is received by PEMP enable router, the corresponding routing game is rebuilt and the routing decision is made in response to the game equilibrium routing. To simulate a real operational router and evaluate its processing time under different traffic load scenarios, we increase the incoming data-plane traffic rate. The stress-test result is presented in Fig. 8, again in terms of average processing time. For this experiment, the processing time is at a much smaller scale than for IGP link state changes (ms instead of s). The difference between standard BGP and PEMP is this time much smaller (lower than $2ms$), and almost negligible for low and medium loads. However, for high loads the processing time gap with PEMP increases to roughly 20%, which is not enormous, also considering that for very high bit rate the usage of open source routers is a seldom choice. The marginal gap in high-end multi-core routers is expected to be much lower.

VI. Conclusions

In this paper we presented how Peering Equilibrium MultiPath (PEMP) routing can be implemented in real routers. PEMP routing was proposed five years ago for making inter-domain routing more stable, in particular across peering settlements among Internet Autonomous Systems.

Its implementation allowed us to validate most of the modeling choices, as well as to revisit some of the design choices at the light of implementation-specific constraints. More precisely, we specified how weighted load-balancing should be performed over PEMP routers, and how equivalent paths can be identified. We also specified how the forwarding logic should operate a dual logic for standard traffic and for PEMP traffic.

By means of extensive tests on realistic emulated network interconnections, we showed that PEMP can be integrated at low computation overhead. We released the PEMP-capable open-source Quagga-based router code [9].

References

[1] M. Caesar, J. Rexford, "BGP routing policies in ISP networks", *IEEE Network* 19(6), 2005.

[2] Y. Rekhter, T. Li, "A Border Gateway Protocol 4 (BGP-4)", RFC 1771, 1995.

[3] R. Teixeira et al., "Impact of Hot-Potato Routing Changes in IP Networks", *IEEE/ACM Trans. on Networking* 16(6), 2008.

[4] R. Mahajan, D. Wetherall, T. Anderson, "Negotiation-based routing between neighboring ISPs", *Proc. of NSDI 2005*.

[5] W. Xu, J. Rexford, "MIRO: Multi-path Interdomain Routing", *ACM SIGCOMM Computer Communications Review* 36(4), 2006.

[6] R. Mahajan, D. Wetherall, T. Anderson, "Towards coordinated interdomain traffic engineering", in *Proc. of HotNets-III 2007*.

[7] R. Mahajan, D. Wetherall, T. Anderson, "Mutually controlled routing with independent ISPs", *Proc. of NSDI 2007*.

[8] S. Secci et al., "Peering Equilibrium MultiPath Routing: a game theory framework for Internet peering settlements", *IEEE/ACM Trans. on Networking* 9(2), 2011.

[9] Routing games LIP6 open source project (website): https://routing-games.lip6.fr.

[10] A. Orda, R. Rom, N. Shimkin, "Competitive routing in multiuser communication networks", *IEEE/ACM Trans. on Networking* 1(5), 1993.

[11] Y. Korilis, A. Lazar, A. Orda, "Architecting noncooperative networks", *IEEE J. on Sel. Areas in Communications* 13(7), 1995.

[12] R. Feldmann et al., "Nashification and the coordination ratio for a selfish routing game", *Automata, Languages and Programming*. Springer Berlin Heidelberg, 2003.

[13] R. Etkin, A. Parekh, D. Tse, "Spectrum sharing for unlicensed bands", *IEEE J. on Selected Areas in Communications* 25(3), 2007.

[14] S. Y. Yun, A. Proutiere, "Distributed Proportional Fair Load Balancing in Heterogenous Systems", *Proc. of ACM SIGMETRICS 2015*.

[15] H. Kameda, E. Altman, "Inefficient noncooperation in networking games of common-pool resources", *IEEE J. on Selected Areas in Communications* 26(7), 2008.

[16] K. Ishiguro, "Quagga Software Routing Suite"(website): http://www.quagga.net.

[17] A. Medina, A. Lakhina, I. Matta, J. Byers, "BRITE: An Approach to Universal Topology Generation", in *Proc. of MASCOTS 2001*.

[18] A.L. Barabasi, R. Albert, "Emergence of Scaling in Random Networks", *Science* 286(5439), 1999.

[19] T. Bu and D. Towsley. "On distinguishing between Internet power law topology generator", in *Proc. of INFOCOM 2002*.

[20] A. Markopoulou et al., "Characterization of Failures in an IP Backbone", *IEEE/ACM Trans. on Networking* 16(4), 2008.