BGP Add-Paths : The Scaling/Performance Tradeoffs

Virginie Van den Schrieck, Pierre Francois, Olivier Bonaventure
Université catholique de Louvain, Belgium

***Abstract*— Internet Service Providers design their network with resiliency in mind, having multiple paths towards external IP subnets available at the borders of their network. However, with the current internal Border Gateway Protocol, BGP routers and route reflectors only propagate their (unique) best path over their iBGP sessions. As a result, at the BGP router level, path diversity tends to be poor. Such lack of path diversity can lead to MED oscillations, prevents an efficient use of multipath BGP and does not allow for a fast and local recovery upon nexthop failure. Advertising multiple paths over iBGP sessions with BGP *Add-Paths* solves those issues, depending on the way the additional paths are selected. In this paper, we analyze the various options for the selection mode of the paths to be advertised. We show that these modes differently fulfill the needs of *Add-Paths* applications such as fast recovery upon failure and MED oscillation avoidance. We also show in our analysis that the costs and benefits bound with these modes depend on the connectivity of the AS where it is deployed. To support the analysis, we developed a tool allowing to measure the scaling of these modes in a given network. We illustrate the utilization of this tool on synthetic Internet topologies, and provide some recommendations for the choice of an *Add-Paths* selection mode.**

I. Introduction

BGP [1] is the interdomain routing protocol that distributes reachability of IP subnets in the Internet. External BGP (eBGP) sessions are used among adjacent routers belonging to different Autonomous Systems (ASes) to exchange paths, while internal BGP (iBGP) sessions are used among routers belonging to the same AS to exchange the paths learned at the border of their own AS.

For scaling and ease of management purposes, many ISPs have moved their iBGP architecture from a full mesh of iBGP sessions to Route Reflection [2]. With Route Reflection, some selected routers, called Route Reflectors, collect paths received from their connected AS Border Routers (ASBRs). A Route Reflectors advertises its best path to other Route Reflectors as well as to its connected ASBRs. ASBRs are typically connected to two Route Reflectors in their AS. Multi-level Route Reflection is sometimes used by very large ISPs. In such a case, Route Reflectors are themselves clients of top level Route Reflectors.

In the example of figure 1, $RR1$ and $RR2$ are both Route Reflectors. ASBRs $R1$, $R2$ and $R3$ announce their best path towards each prefix to those two Route Reflectors only.

ISPs usually design their networks with resiliency in mind. They tend to multihome, i.e. connect to multiple providers and peers, and they tend to multi-connect, i.e. they have multiple eBGP links with their neighboring ASes [3][4][5]. Multi-connectivity with the same AS is also often motivated by bandwidth requirements. As a result, multiple in-policy BGP nexthops are most often available for each IP subnet in an AS network [6].

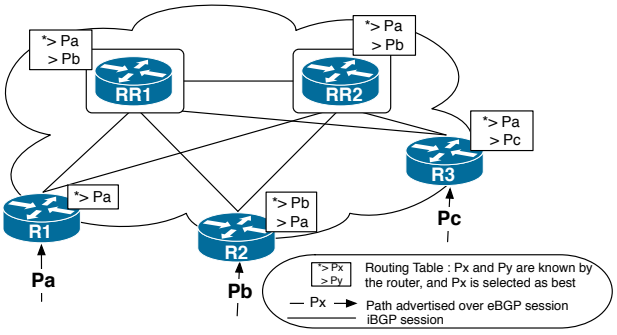

Fig. 1. ISP using Route Reflection

Nevertheless, even though multiple paths towards a single IP prefix are available at the borders of an AS, when looked at the router level, diversity tends to be poor [6]. This is mainly due to two factors. First, Route Reflectors have normal iBGP sessions with their clients, hence they only propagate one best path to their clients. Furthermore, the two Route Reflectors to which an ASBR connects usually pick the same path, which does not help improving the path diversity on ASBRs. Second, ASBRs having learned external routes do not advertise them to their Route Reflectors when they prefer an iBGP learned path over their external ones.

In other words, AS-wide path diversity is usually present for any given prefix at the borders of the AS, but router-local diversity is not necessarily ensured in all current iBGP designs.

In the example of figure 1, three paths to destination d are learned by the ISP : Pa, Pb and Pc. As both Route Reflectors prefer the path Pa, path Pb is not advertised to the other ASBRs. Also, due to policies (ex : lower local preference), router $R3$ does not advertise path Pc as it prefers the iBGP-learned path Pa. As a consequence, Pb and Pc are not advertised to other ASBRs, and router $R1$ only knows about one path.

Such a lack of router-local diversity can prevent fast recovery when a router or peering link fails. For example, it prevents a fast data-plane activation of alternate nexthops, as provided by the Prefix Independent Convergence feature [7]. It also reduces the efficiency of multipath BGP based on BGP nexthop load balancing [8]. Furthermore, hidden paths are the source of iBGP routing oscillations caused by MED [9]. Finally, when a border router fails, an ASBR

which does not yet know its post-convergence path must wait for the subsequent iBGP reconvergence. In the meantime, it can trigger the propagation of transient BGP Withdraw messages over its eBGP sessions, leading to transient losses of connectivity [10][11]. Bursts of transient BGP Updates that will eventually be re-updated with the post-convergence paths may also be leaked out over eBGP sessions. Such a behavior contributes to inter-domain routing churn even in cases where a failure can be handled locally by the ISP.

On Figure 1, upon failure of path Pa, router $R1$ cannot reach destination d anymore and will drop packets towards d until the Route Reflectors advertise Pb. Furthermore, $R1$ will also send eBGP withdraws on its eBGP sessions.

The impact of the lack of router-level path diversity can be mitigated in some network designs, where each PoP area is connected to the core network with a pair of Area Border Routers typically acting as Route Reflectors. As the RRs are on the forwarding paths, they can advertise their "Best PoP" paths to each other and replace the nexthop attribute with their own IP address in order to maintain connectivity in case of failure [12]. Such BGP designs assume a well controlled hierarchical IGP design and route reflector placement in order to work properly.

A solution to solve the lack of router-level path diversity in any network without changing iBGP architectures and operational habits is to allow for the dissemination of BGP nexthop-disjoint paths towards the same IP prefix, over iBGP sessions. This is the goal of *Add-Paths*, a technique undergoing standardization at the IETF [13]. This standardization activity is focused on the encoding and signaling of such multiple paths. However, the selection of the set of paths to be advertised by routers and Route Reflectors is left to the implementations, depending on the application of *Add-Paths* that they want to support. We call the algorithms to select that set of paths *Add-Paths Selection Modes*. In this context, this paper provides ground to vendors for making a decision on which *Add-Paths* selection modes they want to support, and to operators for which *Add-Paths* selection modes they want to request to their vendors, and what will be the expected impact of these choices.

In this paper, we first analyse in section II the properties met by the set of paths advertised by *Add-Paths*. As those properties are dependent on the way those sets of paths are computed, we then present in section III various approaches for selecting the additional paths to be advertised. We analyze how they can fit with applications requiring more router-level path diversity, and at which cost. Some performance criteria of these *Add-Paths selection modes* heavily depend on the deployment scenario, i.e., on the connectivity of the AS where it is deployed. Hence, to support such analysis, we present in section IV a tool allowing us to perform case-by-case studies for ISPs who want to get actual numbers w.r.t their network. We then show an example of the kind of results that can be provided by the tool.

II. Properties of AddPaths selection modes

Depending on how they are selected, the paths advertised by *Add-Paths* have different properties. They will thus unequally meet the different objectives of *Add-Paths* presented in the introduction, and bear different control-plane overheads. In this section, we will discuss the various evaluation criteria of our analysis of Add-path selection modes. We assume that the ISPs use encapsulation (e.g. MPLS) to prevent forwarding loops and path deflection to occur [14]. Most ISPs already use MPLS to support BGP/MPLS VPNs or for traffic engineering purposes.

We discuss in section III the ability of each mode to provide **nexthop-disjoint alternate paths** to each BGP speaker of the AS, provided that such paths are available at the borders of the network. This property ensures that routers will be able to use multipath BGP [8] for load balancing and will be able to use a fast recovery technique such as Prefix Independent Convergence [7] in case of peering links or border routers failures.

We also discuss the ability of each mode to **avoid MED oscillations** [9]. This was the first motivation for *Add-Paths*. Some of the modes will indeed allow routers to learn paths that would have otherwise been hidden to them. This increased diversity can result in a guaranteed iBGP convergence.

Route Reflection is known to sometimes provide sub-optimal routing because route reflectors perform an IGP tie-break based on their own IGP distances, which may differ from the IGP tie-break that the client they serve would perform. By advertising additional paths with *Add-Paths*, **optimal routing** can be re-ensured if the best paths from the perspective of these clients are advertised to them.

By increasing the router-level BGP diversity within an AS, *Add-Paths* reduces the likelihood of propagation of bursts of BGP Withdraw and Update messages outside the AS for a given prefix, which can occur during a BGP convergence following a local link or router failure [10][11]. Indeed, with *Add-Paths*, BGP routers are more likely to already know their post-convergence paths at the time of the convergence. We will discuss this under the term **eBGP churn reduction**.

The cost of providing such diversity can also vary among the path selection modes. Providing additional paths over iBGP sessions comes at the cost of reflecting their updates and re-triggering the BGP decision process more often, instead of keeping the paths hidden at the borders of the AS. That is, the eBGP churn reduction discussed above comes at the cost of an increase of the iBGP churn on non-best paths. Note that this cost is only related to the control-plane, as updates on non-best paths do not impact the FIB of routers. We will term this evaluation criterion **Control-plane stress**.

The Adj-Rib-Ins of BGP routers will contain more paths and thus use more memory than without *Add-Paths*. We will analyze this memory increase under the term **Control-plane load**. Note that the actual memory increase due to the reception of more paths towards the same IP subnet is rendered sub-linear with the number of additional paths thanks to attribute-sharing.

Some *Add-Paths* selection modes might require more CPU cycles than others for selecting paths. In some circumstances, Adj-Rib-In optimizations can make such decisions trivial, while for others the algorithm is more complex. We will term this criterium as **Decision Process Complexity**.

III. ADD-PATHS SELECTION MODES

In this section, we review the main *Add-Paths* selection modes that are considered for deployment. Due to space limitations, we cannot provide a detailed analysis for all of them. More details can be found in [15].

A. Add-All-Paths

A simple rule for advertising multiple paths in iBGP is to advertise to iBGP peers all received paths, provided they respect export rules such as cluster-id checks.

This solution gives a perfect path visibility to all routers, thus limiting at best the eBGP churn and transient losses of connectivity in case of nexthop failure, and provides all the paths that a router may consider for actual use with multipath BGP. As no paths are hidden from any BGP router, MED oscillations cannot occur with *Add-All-Paths*. Also, as no local decision is made by Route Reflectors to not propagate paths to their clients, these have full knowledge of paths and can pick the optimal (hot-potato) one w.r.t. their own IGP distances.

As all paths are known by each BGP router, the post convergence path following an internal event like an IGP event or the loss of the BGP nexthop is already available to the routers that will perform rerouting w.r.t. this event. As a result, the sending of BGP updates over eBGP sessions will be reduced to its minimum, being the update of the initial path to the post-convergence path.

Add-All-Paths is easy to implement, as all paths are eligible for propagation. The counter part is that all paths need to be stored by all routers, which can consume lots of memory. If a path to a prefix P is advertised to N border routers, with a Full Mesh of iBGP sessions, all routers have N paths in their Adj-Rib-Ins. If *Add-All-Paths* along with Route Reflection is used and each client is connected to 2 Route Reflectors, it may learn up to 2*N paths, as both Route Reflectors will send the full set of available paths. The number of BGP messages disseminated in iBGP is also the worst possible with *Add-All-Paths*.

B. Add-N-Paths

Add-N-Paths is an intuitive selection mode for *Add-Paths*, as it basically provides a configured upper bound N on the number of paths that BGP routers advertise over a single iBGP session. In this paper, we consider an implementation where the selection of these N paths is equivalent to the one obtained by a BGP router which first picks its best path, removes all paths with the same nexthop as the best from its Adj-Rib-Ins, picks its second best on the resulting set of paths, and repeats that process until the resulting set becomes empty or N paths have been picked.

Add-N-Paths with $N = 2$ is a very appropriate mode to enable fast recovery with Prefix Independent Convergence [7] as it ensures the availability of at least 2 nexthop-disjoint paths in any BGP router of the AS, provided that there are at least two paths available at the borders of the network. This mode allows for multipath BGP for the same reasons.

From a theoretical point of view, *Add-N-Paths* could be considered as a bad option because it does not provide guarantees in many aspects. First, *Add-N-Paths* does not guarantee that MED oscillations will be avoided when enabled. Under some circumstances, it is even possible that enabling *Add-N-Paths* leaves the iBGP system in a persistent oscillation in the propagation of non-best paths, although iBGP routing was stable without *Add-Paths*. Examples of such oscillations can be found in [16]. Second, routing optimality is not guaranteed but is more likely to be obtained when N is high. Third, even though an ASBR will learn alternate paths towards all prefixes when available, there is no guarantee that it will know the post-convergence path w.r.t. the convergence event. eBGP churn after a local failure may be reduced, but is not necessarily minimized.

Nevertheless, the load and control-plane stress on the routers can be easily predicted by an ISP, as it is for each router a direct function of the number of iBGP sessions that it maintains, the number of prefixes advertised through the ISP, and the value of N.

The decision process complexity is also related to the value of N, as N runs of decision process are needed to select the paths.

For ISPs who want to achieve fast recovery and easily predict the overhead on the control-plane of its BGP routers, *Add-N-Paths* with a small value for N is likely to be the best option.

C. Add-Group-Best-Paths

The main objective of *Add-Group-Best-Paths* [17] is to avoid MED oscillations. The idea of this mode is to let BGP routers advertise over iBGP the best path that they know for each neighboring AS. As a result, the lowest-MED paths from each neighboring AS are known to all BGP routers, hence non-lowest MED paths cannot be picked as best, guaranteeing convergence. IGP topology-related oscillations [14] are not prevented by this mode, except if some design constraints on the IGP topology are followed.

This mode provides mitigated benefits for applications other than MED oscillations prevention. It could be deployed as an emergency mechanism to be used when MED oscillations are detected on a prefix, as mentioned in Section V.

Regarding fast recovery and load balancing, *Add-Group-Best-Paths* provides one path for each neighboring AS, but not necessarily the post-convergence ones or the optimal ones. The eBGP churn upon primary path failure with this mode will be reduced only if more than one path is propagated, i.e. if the prefix is advertised to the AS by more than one neighbor. However, if the post-convergence path is from the same AS as the primary path, unnecessary BGP updates will be advertised outside the AS. If only one AS advertises some paths towards a prefix, it is even worse, as only one path is propagated.

The increase in control plane stress highly depends on the connectivity of the AS. Large transit ISPs receiving paths towards the same IP prefix from many different ASes will need to store and update one best path per such neighboring AS. ISPs with few different neighboring ASes will not see a large amount of additional BGP Updates flowing through their iBGP architecture.

The decision process for *Add-Group-Best-Paths* is relatively simple. The Adj-Rib-In can be optimized by splitting the set

of BGP paths according to the neighboring AS from which it was received. The decision process then becomes the usual BGP decision process applied on each of these sets. Upon reception of an update, a decision is only to be remade on the subset of paths that corresponds to the neighboring AS from which the update was received.

D. Add-AS-Wide-Bests-Paths

Another solution focused on the avoidance of MED oscillations has been proposed in [9]. The solution avoids MED oscillations by design, letting all BGP routers advertise the paths that remain before applying the IGP tie-break rule. Thus, all paths with the highest local preference, shortest AS path length, and lowest MED value per neighboring AS are eligible for propagation. As a result, a router will eventually know all these paths and will no longer select as best a path with a non-lowest MED attribute. This solution also prevents IGP-topology related oscillations without constraints on the IGP topology.

Enabling this mode as a default choice could prevent fast recovery in the case where only one path meets the selection criteria. This happens when the decisive rule of the BGP decision process is either local preference, shortest AS-Paths or lower MED (among paths from same neighbors). For example, if a prefix is learned on one eBGP session from a peer and two eBGP sessions with providers, only the single path from the peer is propagated to the ASBRs. Fast recovery upon link failure cannot be ensured in this case.

Similarly, with *Add-AS-Wide-Bests-Paths*, the application of multipath BGP is restricted to the cases where multiple paths with the highest local preference, the shortest AS path, and the lowest MED value (per neighboring AS) are available. Note however that this restriction is not considered as an issue as this constraint is the usual policy for multipath BGP applications [8].

Optimal routing is ensured with *Add-AS-Wide-Bests-Paths* as no BGP router prevents itself from advertising some paths based on local decisions.

The computational cost to run this *Add-Paths* selection mode remains low, as compared to vanilla BGP, as it is just no going through the whole sequence of rules that vanilla BGP applies.

The control-plane stress and load increase bound with this mode relates to the amount of equally preferred best paths that are available to the AS. For example, a large transit AS with tens of equally preferred peer paths available for a given prefix will see its BGP control-plane stress and load increased a lot as compared to those ASes who have only a few equally preferred provider paths for most of the Internet prefixes, and many paths for only a bunch of peer and customer prefixes.

E. Add-LP1-LP2-Paths

The last mode discussed in this paper is called *Add-LP1-LP2-Paths*. Its goal is to provide guaranteed fast recovery in case of local failures, load-balancing, MED oscillation avoidance and eBGP churn reduction, with a very simple decision process. It also reduces the control-plane churn and load compared with *Add-All-Paths*.

The idea underlying this mode is to let all the paths with the highest local-preference value be known by all BGP routers in the network. In the case where only one of such paths exists, i.e. there is only one BGP nexthop providing a path with the highest local-preference value, then all the paths with the second highest local-preference value must be announced in BGP as well.

By definition, the post-convergence path following the loss of a primary path belongs to the set of paths with the highest local preference value when more than one such path exist. The post-convergence paths belongs to the set of paths with the second highest local preference value in the other case. Thus, with *Add-LP1-LP2-Paths*, all BGP routers will know about alternate paths, and these contain the post-convergence paths such that the eBGP churn during convergence is minimized.

Add-LP1-LP2-Paths allows for multipath Load Balancing with default policies where paths with the highest local-preference value are eligible.

MED oscillations can only occur among paths having the highest local-preference value, when some of them are kept hidden from some BGP routers. As *Add-LP1-LP2-Paths* enforces the propagation of all the paths with the highest local-preference, MED oscillation cannot happen with this mode.

The Adj-Rib-In can be organized for an optimized support of *Add-LP1-LP2-Paths*. For each IP prefix, the optimized Adj-Rib-In maintains 3 sets of paths. The first set ($LP1$) contains references to the paths having the highest local-preference. The second set ($LP2$) contains references to the paths having the second highest local-preference. The third set contains all the remaining paths. The algorithm to support *Add-LP1-LP2-Paths* selects for advertisement the paths that belong to $LP1$. If $LP1$ only contains one path, it also selects the paths that belong to $LP2$.

The control plane stress and load bound with this solution depends on the number of paths with the highest local preference that an ISP learns at its borders. The more prefixes having at least two paths with the highest preference, the lowest the control plane stress is, as only the paths from $LP1$ needs to be advertised.

F. Summary

Table I summarizes the characteristics of the five selection modes. It shows that both *Add-Group-Best-Paths* and *Add-AS-Wide-Best-Paths* are dedicated to MED oscillation prevention and cannot guarantee the existence of at least one alternate path for each prefix. On the contrary, *Add-N-Paths* modes reduce the likelihood but do not prevent MED oscillations. However, they enable fast recovery and limit churn propagation, with bounded costs. *Add-All-Paths* and *Add-LP1-LP2-Paths* both prevent MED oscillations and enable fast recovery, eBGP churn reduction and path diversity for multipath. Compared to *Add-All-Paths*, *Add-LP1-LP2-Paths* has a lower control plane cost as not all paths are propagated, without losing any of the benefits brought by *Add-All-Paths*.

Note that any Add-Path mode that is constrained with an upper bound on the number of paths that can be advertised for

	Path optimality	Backup path availability/ optimality	Control plane load and stress	DP complexity	MED oscillation avoidance
Add-All-Paths	OK	OK/OK	Max	Easiest	OK
Add-N-Paths	Improved	OK/Improved	Bounded	Hard (related to N)	KO
Add-LP1-LP2-Paths	OK	OK/OK	Max	Easier	OK
Add-Group-Best-Paths	KO	KO/KO	Max	Easy	OK
Add-AS-Wide-Best-Paths	OK	KO/*OK*	Max	Easy	OK

TABLE I
SUMMARY OF SELECTION MODES CHARACTERISTICS

a given prefix has the same limitations as Add-N-Paths w.r.t. MED oscillation avoidance. This is for example the case with *Add-AS-Wide-Best-Paths* limited to a maximum of 5 paths.

Whether *Add-N-Paths*, *Add-LP1-LP2-Paths* or *Add-All-Paths* should be preferred when fast recovery and multipath are the target *Add-Paths* applications depends on the network connectivity. It depends on the resources available in the network as well as on its topology and the way it interconnects with other ASes. The tool presented in section IV allows to quantitatively compare the cost of each selection mode on a given network.

IV. EVALUATION OF ADD-PATHS SELECTION MODES

Some characteristics of the selection modes proposed in section III are highly network dependent. For example, the memory load of a mode such as *Add-LP1-LP2-Paths* depends on the number of paths having the highest preference, which in turns depends on the policies used by the ISP and on the peerings between the ISP and its neighbors. It might be difficult for a provider to correctly evaluate a-priori the costs and benefits of using a given selection mode. Thus, we have developed a tool based on simulations to performs such evaluation. The operators can then evaluate the topology-dependent tradeoffs of each selection mode when applied on their network. In this section, we first present the tool, then we illustrate its usage in two cases, with synthetic Internet topologies.

A. Evaluation tool

The tool is built around SimBGP, a BGP simulator written in Python [18]. SimBGP is well suited for dynamic BGP simulations, as the propagation and processing delays of the messages are taken into account. It is an event-driven simulator that relies on an ordered queue to successively process simulation events. The simulator has been previously used to evaluate solutions aimed at improving the resiliency of interdomain routing [19][20]. The original simulator supports the classical BGP decision process as well as a generic multiple paths advertisement. We extended it[1] to support the *Add-Paths* encoding and all the selection modes presented in section III. We also added a more detailled IGP layer to offer a better support of Hot Potato routing.

The evaluation tool handles the setup of the SimBGP simulation with a given topology and the selected selection mode. It initiates BGP events such as path advertisements or peering link failures and monitors the BGP convergence and the resulting routing tables to extract a set of metrics, which can be targeted either on the provider under test or on all ASes of the simulated topology. Our first metric is the **dataplane convergence time**, defined as the time after which no router lacks reachability in the dataplane for the destination. A related metric is the **control plane convergence time**. This is the time between the beginning of an event and the moment where the last BGP message triggered by the event is received. The difference between the impact on the dataplane and the impact on the control plane of a selection mode can also be measured in terms of the number of BGP messages exchanged, by comparing the **total number of BGP messages** exchanged during the simulation versus the **number of important BGP messages**, which is the number of messages that modify the best path selection of a router (i.e. new best path, best path removed or best path changed). Finally, our last metric measures the **impact of an event**, i.e. the percentage of ASes that receive BGP messages about an event.

B. Generation of Internet topologies

The tool can be used by operators to evaluate the cost of deploying *Add-Paths* in their network, but it can also be used on synthetic topologies to provide an indicative evaluation of the selection modes. The remainder of this section presents a study of the selection modes based on synthetic Internet-like topologies. The study focuses on two scenarii : ISPs receiving two paths for a prefix from a dual-connected stub, and ISPs receiving multiple paths for a prefix from their peers/providers. We first present the topologies used in the study, then we detail each scenario in the next subsections.

Our topology generation methodology is a two-steps process : First, we use Ghitle [21] to generate an AS-level topology that includes the business relationships between ASes, then we iterate on all ASes to define the internal structure of each of them as well as the way it interconnects on the router-level with its neighbors. Once the AS-level topology has been defined, we generate the internal structure of each AS : Number of routers, IGP topology, iBGP topology and eBGP connectivity by using iGen [22]. In order to build an IGP topology, IGen first places routers at random location, then groups the routers in clusters/PoPs. We choose a cluster size of 10 routers. Among each cluster, two routers are choosen to form the backbone and connect with other clusters. Other routers of the cluster have links with both backbone routers. IGP links are assigned based on the geographical distances

[1]The modified simulator and all validation tests are available at http://inl.info.ucl.ac.be/softwares/simbgp-addpaths-support

between routers. IGen then build the iBGP topology of the domain on top of the IGP topology : Backbone routers are the Route Reflectors, and other routers connect to the two Route Reflectors of their PoPs.

Domains at each level of the topology share the same parameters, summarized in table II. The number of peering links between two domains is proportional to the number of routers in both domains.

We built 10 topologies with those parameters. The resulting SimBGP configuration files are available online [23].

Level	#Num ASes	# routers per AS	iBGP org.
Tier-1	[10-15]	[100-150]	Redundant RRs
Tier-2	[50-70]	[10-50]	Redundant RRs
Tier-3	[100-200]	[10-15]	Redundant RRs

TABLE II
PARAMETERS OF A DOMAIN TOPOLOGY

C. Scenario 1 : Dual-connected stubs

In this scenario, we randomly pick a total of 90 providers among the 10 synthetic Internet topologies. For each of those providers, we select up to 20 pairs of routers and connect them with a dual-connected stub advertising a prefix, as shown in figure 2. The provider is configured to use *Add-Paths*, while the other ASes run vanilla BGP. We parametrized the simulator with no MRAI timer and a processing time of BGP messages between 1 and 10 milliseconds. We then use our tool to compute the metrics during prefix advertisement, then upon failure of one link between the ISP and the stub. For each metric, we compute the ratio between the value for each selection mode and the value for vanilla BGP. The modified metric value represents the gain/overhead of *Add-Paths* compared to vanilla BGP. The value of each modified metric for vanilla BGP is thus always one. We then compute the means of each value. The 95th confidence interval of each mean is below 3% of the mean value.

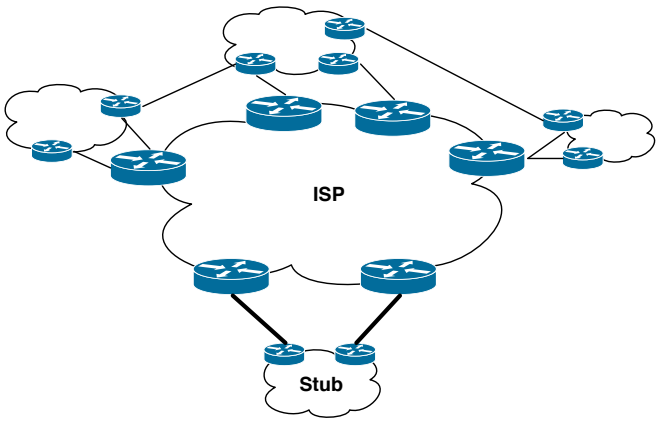

Fig. 2. Stub dual-connected to its provider

Having a stub advertising a prefix on two links means that the AS knows two equivalent paths (same local preference in the simulation) to that prefix. With vanilla BGP, as Route Reflectors advertise only their best path to their clients and Route Reflectors of the same cluster are likely to select the same path as best, some routers only learn one path. This problem is studied in details in [6].

With *Add-N-Paths*, *Add-All-Paths*, *Add-LP1-LP2-Paths*, backup paths are available by design, while with *Add-AS-Wide-Bests-Paths*, both paths are known because they have the same local preference. The metric measuring the number of paths confirms this, as the average number of paths is twice higher with *Add-Paths* than with vanilla BGP. *Add-Group-Best-Paths* does not enforce backup path availability because both available paths come from the same AS, and the control plane load is thus similar to the one with vanilla BGP.

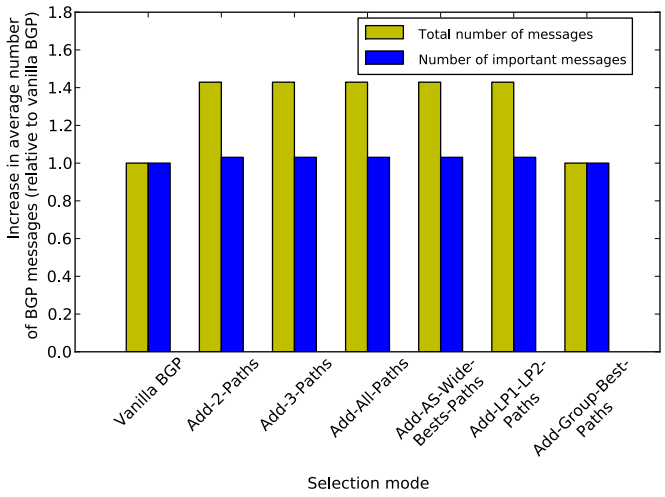

Fig. 3. Increase in the number of BGP messages exchanged in the provider upon advertisement of a prefix by the dual-connected stub

Upon initial advertisement of the prefix, a control plane stress overhead is encountered with the four modes providing backup paths, as more paths are exchanged inside the ISP. We measured in our simulations 1.4 times more BGP messages with those modes than with vanilla BGP and *Add-Group-Best-Paths*, as shown in figure 3. However, the number of important messages, i.e. those that change the best path of the router, is roughly the same in all modes. The overhead of using *Add-Paths* upon prefix advertisement is thus mainly due to the exchange of additional paths, and the best path selection is not impacted. This is confirmed by the dataplane convergence time, which is identical in all modes. *Add-Paths* does not delay the reachability of a new prefix.

Once the prefix is known in the whole topology, we successively fail both links between the stub and the ISP.

When a link fails, with vanilla BGP and *Add-Group-Best-Paths*, all routers that use the path via this link and do not know an alternate path will encounter dataplane disruptions and send BGP messages outside the AS. Other ASes will then be impacted by the event. With all *Add-Paths* modes except *Add-Group-Best-Paths*, all routers of the provider know both paths. The link failure can be recovered immediately and locally. Figure 4 shows that the value of the metric measuring the BGP dataplane convergence time is 0 when backup paths are available. This is also illustrated by the

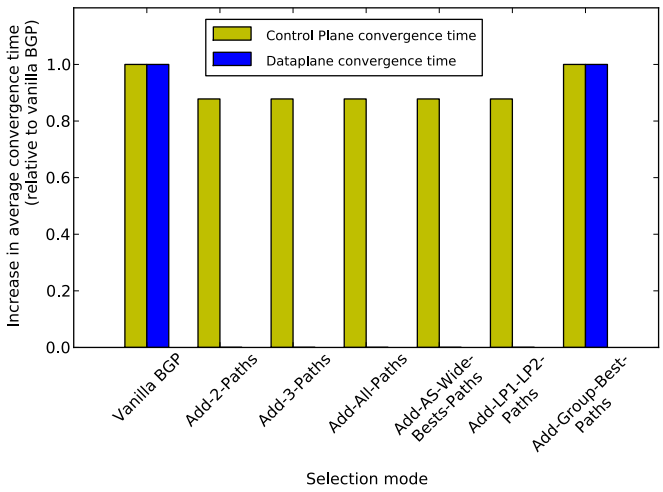

Fig. 4. Increase in Dataplane and Control Plane convergence times upon link failure recovery

metric measuring the percentage of external ASes impacted : On average, **with vanilla BGP, 12,5% of the ASes in our synthetic Internet learn about the failure, while with *Add-Paths*, no other AS than the provider processes BGP messages about the failure**. *Add-Group-Best-Paths* has the same behaviour as vanilla BGP, as it does not provide an alternate path in this case. This confirms the adequacy of the *Add-Paths* selection modes advertising at least two paths for providing fast recovery upon link failure.

On the control plane side, *Add-Paths* is also slightly quicker than vanilla BGP when backup paths are available (about 10% in figure 4), as only BGP messages about the failure need to be propagated in the AS versus BGP messages about both the failed path and the backup path.

D. Scenario 2 : Prefixes advertised from other providers/peers

Scenario 1 shows how Add-Paths can provide fast and local recovery upon stub link failure when the selection mode allows the propagation of at least two different paths. However, a typical ISP is of course also connected to non-stub ISPs from which it receives Internet destinations. Similarly to the first scenario, fast and local recovery is ensured upon failure of a link to those neighbors with the proper Add-Paths mode, as long as there is an alternate path available in the AS. This is often the case as those neighbors are likely to be more than dual-connected and Internet destinations are possibly reachable via several neighbors, but the related cost of deploying Add-Paths is probably higher than for dual-connected stubs. In this second scenario, we will thus evaluate the cost of a full Add-Paths deployment allowing fast and local recovery, load balancing and/or MED oscillation prevention.

For this evaluation, we take our 90 providers and for each of them, we let 20 single-connected stubs randomly located in the corresponding Internet topology advertise one prefix each. The provider under test will thus learn different paths for each of these prefixes depending on its peerings with other ISPs. On the example of figure 2, the provider might learn about each prefix from its three neighbors, depending on its policies. If all neighbors advertise the prefix, it will learn up to 6 paths.

For each prefix, we compute our metrics to show the additional load and resulting diversity on the provider under test. The results presented here are the means of the metric values for each prefix. We also classify the ISPs under test depending on the level of the topology to which they belong : Tier-1 or Transit ISP (Tier-2 or Tier-3).

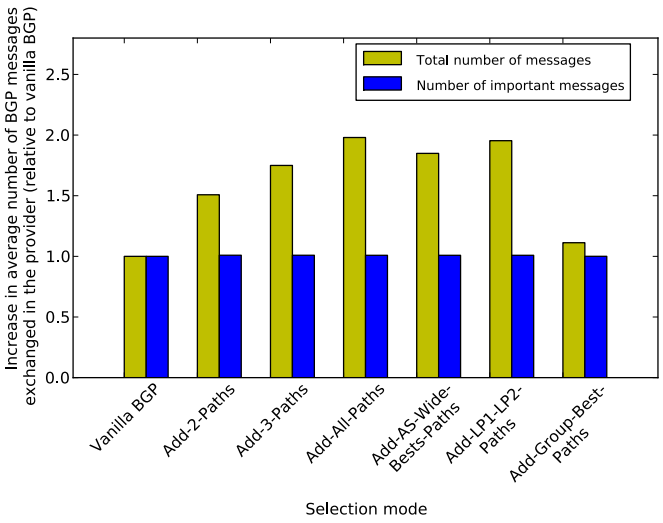

Fig. 5. Increase in the number of messages exchanged for a prefix inside a Transit ISP

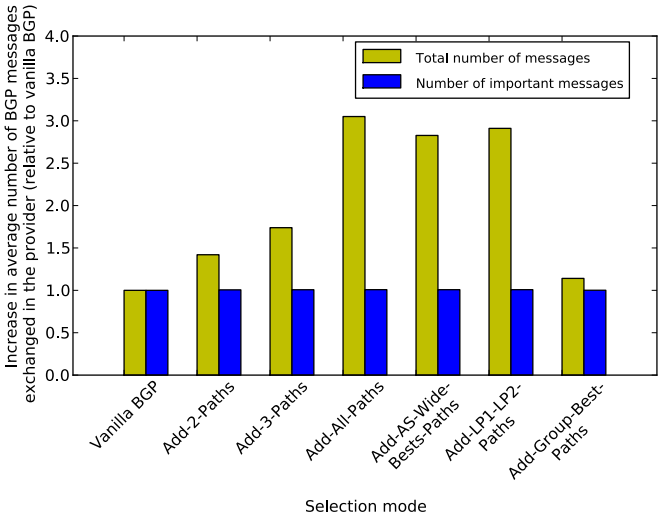

Fig. 6. Increase in the number of BGP messages exchanged for a prefix inside a T1 ISP

Figure 5 shows the average number of BGP messages exchanged for a prefix inside Transit ISPs, while figure 6 shows the same metrics for T1 ISPS. Similarly to what was observed in the case of the dual-connected stub, advertising the additional paths increases the number of BGP messages exchanged inside the AS, but the number of messages impacting the best path selection remains stable. The increase in the number of BGP messages also varies depending on the selection modes. Modes with a bounded number of paths

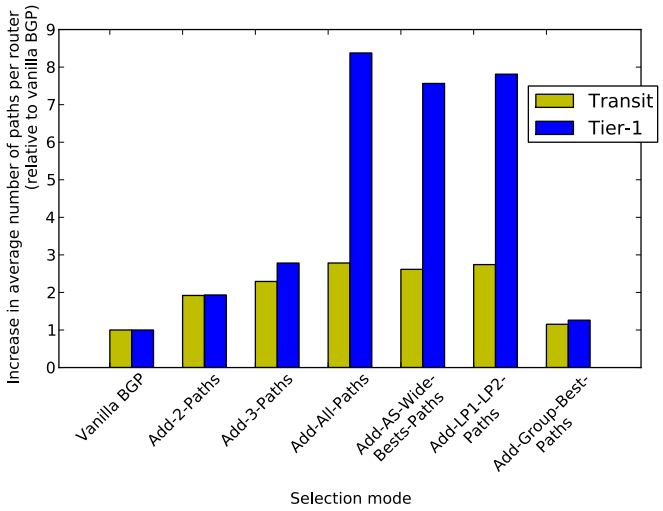

Fig. 7. Increase in the mean number of paths for a prefix learned by a router

have of course a limited control plane stress, while the impact of the other modes depends on the topology. For example, routers of a T1 ISP exchange up to 3 times more messages with those modes while routers of smaller Transit ISPs exchange twice more messages than with vanilla BGP. The *Add-Group-Best-Paths* selection mode has the smallest impact among all modes, and is only slightly more costly than vanilla BGP. This is because in our topologies, prefixes are mostly learned on multiple sessions with a single neighbor.

The memory cost of each selection mode is shown on figure 7. Roughly, the increase in terms of control plane load when using *Add-N-Paths* is proportional to the number of paths disseminated, whatever the level to which the ISP belongs. The memory load is bounded by N. However, with *Add-All-Paths*, *Add-AS-Wide-Bests-Paths* and *Add-LP1-LP2-Paths*, the number of paths is not bounded, and depends on the number of paths available in the AS. This number of available paths depends itself on the level to which the ISP belongs : Large, highly connected ISPs will have more paths than small providers with a few peering/provider links. In our topologies, routers of T1 ISPs learn on average 9 times more paths than with vanilla BGP, while routers of smaller Transit ISPs learn between 2 and 3 times more paths than with vanilla BGP. We can also notice that on T1 ASes, it is slightly less costly to use *Add-AS-Wide-Best-Paths* or *Add-LP1-LP2-Paths* than *Add-All-Paths*. This is because, among the set of received paths, a few of them have a lower local preference and are thus not advertised by the last two modes. Similarly to what was observed for the control plane stress, *Add-Group-Best-Paths* has a control plane load very similar to vanilla BGP. Such a result would encourage an operator only wishing to prevent MED oscillations to use this mode, provided that its IGP topology meets the constraints specified in [17]. Otherwise, he should rather use another mode like *Add-AS-Wide-Bests-Paths*, at the cost of a higher control plane stress and load.

E. Conclusion of the evaluation

This analysis illustrates the kind of conclusions that can be drawn by using our tool. Those results are dependent on the ISP under test, however, we can still observe some interesting generic results about *Add-Paths*. First, using *Add-Paths* has no negative impact on the dataplane during normal operation. The overhead is mainly caused by the exchange of additional paths, which is a control-plane issue. Furthermore, after a link failure, *Add-Paths* allows for a faster recovery as all routers can use a backup path as soon as they learn about the failure. The second scenario of simulations also shows the overhead of the different selection modes in terms of control plane stress and load. We saw that this overhead depends on the size of the AS and on the number of interconnections with other ASes : A relatively small ISP can probably afford chatty modes like *Add-All-Paths* or *Add-LP1-LP2-Paths*, while larger ISPs could prefer using bounded modes like *Add-N-Paths*, depending on the application for which they enable *Add-Paths*.

V. Deployment options

As the BGP protocol is modified by *Add-Paths*, routers need to be upgraded in order to benefit from this new feature. Whether *Add-Paths* is to be deployed on all routers or on a subset of these is an operational choice. Also, all routers do not necessarily need the same path selection mode, depending on their needs and on their available resources.

Different deployment schemes could be imagined : Deploy *Add-Paths* on Route Reflectors only with *Best-External* enabled on ASBRs [24], deploy *Add-Paths* for a given AFI/SAFI (for example, for Internet routes or for VPNs), or even for specific prefixes matching a given access control list (for example, *Add-Group-Best-Paths* for oscillating prefixes). The implications of each deployment should be investigated further. In particular, we do not know yet how different selection modes might interact with each other in an heterogenous deployment.

One can also imagine to deploy *Add-Paths* on Route Reflectors that are off-paths, i.e. that do not forward packet and are only dedicated to distributing paths for ASBRs. Such a solution can be useful if it appears that the processing stress for computing and processing additional paths has an impact on the dataplane performances.

VI. Conclusion

In this paper, we have provided a detailed, qualitative analysis of how to select paths when advertising multiple paths over iBGP sessions with *Add-Paths*. For each mode, we have listed the applications for which they are suited as well as the related cost. We have also presented a tool that allows operators to quantitatively evaluate the cost of deploying each *Add-Paths* selection mode. An application of this tool on synthetic topologies has confirmed our qualitative analysis of the different modes, and highlighted the fact that the modes with non-bounded number of paths behave differently on small or large networks. Such modes can probably be deployed safely on small ISPs, while larger ISPs might need to verify the adequacy of their routers with the memory requirements

of these modes. If MED oscillation prevention is not the goal, an alternative might be to use cost-bounded modes with fixed number of advertised paths.

Further investigation is however needed to study the implications of the different deployment schemes that can be imagined. Also, other selection modes could be explored, such as modes allowing to disseminate paths sharing some characteristics with the primary paths, or modes allowing to select paths based on the communities associated with those paths.

ACKNOWLEDGEMENTS

This work was partially supported by the European-funded Trilogy project and Cisco Research. Pierre Francois's work is supported by the "Fonds National de la Recherche Scientifique", Belgium. We would also like to thanks Clarence Filsfils and Pradosh Mohapatra for their valuable comments.

REFERENCES

[1] Y. Rekhter, T. Li, and S. Hares, "A Border Gateway Protocol 4 (BGP-4)," January 2006, internet RFC4271.

[2] T. Bates, R. Chandra, and E. Chen, "BGP route reflection - an alternative to full mesh iBGP," April 2000, internet RFC 2796.

[3] P. Merindol, V. V. den Schrieck, B. Donnet, O. Bonaventure, and J.-J. Pansiot, "Quantifying ASes Multiconnectivity using Multicast Information," in *Proc. ACM USENIX Internet Measurement Conference (IMC)*, November 2009.

[4] N. Feamster, Z. M. Mao, and J. Rexford, "BorderGuard: Detecting Cold Potatoes from Peers," in *Internet Measurement Conference*, Taormina, Italy, October 2004.

[5] B. Augustin, B. Krishnamurthy, and W. Willinger, "Ixps: mapped?" in *IMC '09: Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference*. New York, NY, USA: ACM, 2009, pp. 336–349.

[6] S. Uhlig and S. Tandel, "Quantifying the impact of route-reflection on BGP routes diversity inside a tier-1 network," in *IFIP Networking 2006*, Coimbra, Portugal, May 2006.

[7] C. Filsfils, "BGP convergence in much less than a second," Presentation at Nanog 40, June 2007. [Online]. Available: http://www.nanog.org/meetings/nanog40/presentations/ClarenceFilsfils-BGP.pdf

[8] "BGP Multipath," Cisco online documention. [Online]. Available: http://www.cisco.com/en/US/tech/tk365/technologies_tech_note09186a0080094431.shtml#bgpmpath

[9] A. Basu, C.-H. L. Ong, A. Rasala, F. B. Shepherd, and G. Wilfong, "Route oscillations in I-BGP with route reflection," in *SIGCOMM '02*. ACM, 2002.

[10] 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 *SIGCOMM '06*. New York, NY, USA: ACM, 2006, pp. 375–386.

[11] V. V. den Schrieck, P. Francois, C. Pelsser, and O. Bonaventure, "Preventing the Unnecessary Propagation of BGP Withdraws," in *Proceedings of IFIP Networking*, 2009.

[12] R. Raszuk, "To Add-Paths or not to Add-Paths," February 2010, NANOG 48 - http://www.nanog.org/meetings/nanog48/presentations/Tuesday/Raszuk_To_AddPaths_N48.pdf.

[13] D. Walton, A. Retana, E. Chen, and J. Scudder, "Advertisement of Multiple Paths in BGP," 2010, internet draft, draft-ietf-idr-add-paths-03.txt, work in progress.

[14] T. Griffin and G. Wilfong, "On the correctnes of iBGP configuration," in *SIGCOMM'02*, Pittsburgh, PA, USA, August 2002, pp. 17–29.

[15] V. V. den Schrieck and P. Francois, "Analysis of paths selection modes for add-paths," Internet draft draft-vvds-add-paths-analysis-00, July 2009.

[16] V. V. den Schrieck and O. Bonaventure, "Routing oscillations using BGP multiple paths advertisement," UCL - IP Networking Lab, Tech. Rep., 2007. [Online]. Available: http://inl.info.ucl.ac.be/publications/routing-oscillations-using-bgp-multip

[17] E. Chen and N. Shen, "Advertisement of the group best paths in BGP," Internet draft draft-chen-bgp-group-path-update-02, September 2004.

[18] J. Qiu. SimBGP : Python Event-driven BGP simulator. [Online]. Available: http://www.bgpvista.com/simbgp.php

[19] F. Wang and L. Gao, "Path diversity aware interdomain routing," in *INFOCOM 2009, IEEE*, April 2009, pp. 307–315.

[20] ——, "A Backup Route Aware Routing Protocol - Fast Recovery from Transient Routing Failures," *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*, pp. 2333–2341, 13-18 April 2008.

[21] C. Delaunois. Ghitle : Generator of Hierarchical Internet Topologies using LEvels. [Online]. Available: http://ghitle.info.ucl.ac.be/

[22] B. Quoitin, V. V. den Schrieck, P. Francois, and O. Bonaventure, "IGen: Generation of Router-level Internet Topologies through Network Design Heuristics," in *Proceedings of the 21st International Teletraffic Congress*, September 2009.

[23] "Topologies used for simulations." [Online]. Available: http://inl.info.ucl.ac.be/softwares/simbgp-addpaths-support

[24] P. Marques, R. Fernando, E. Chen, and P. Mohapatra, "Advertisement of the best-external route to iBGP," July 2008, internet Draft, draft-marques-idr-best-external-00, work in progess.