This document is published in:

Computer Networks, 2013, 57(4), 954-975.
DOI: http://dx.doi.org/10.1016/j.comnet.2012.11.011

BGP-XM: BGP eXtended Multipath for Transit Autonomous Systems

Jose M. Camacho, Alberto García-Martínez, Marcelo Bagnulo, Francisco Valera

Universidad Carlos III de Madrid, Av. Universidad 30, 28916 Leganes (Madrid), Spain

Abstract

Multipath interdomain routing has been proposed to enable flexible traffic engineering for transit Autonomos Systems (ASes). Yet, there is a lack of solutions providing maximal path diversity and backwards compatibility at the same time. The BGP-XM (Border Gateway Protocol-eXtended Multipath) extension presented in this paper is a complete and flexible approach to solve many of the limitations of previous BGP multipath solutions. ASes can benefit from multipath capabilities starting with a single upgraded router, and without any coordination with other ASes. BGP-XM defines an algorithm to merge into regular BGP updates information from paths which may even traverse different ASes. This algorithm can be combined with different multipath selection algorithms, such as the K-BESTRO (K-Best Route Optimizer) tunable selection algorithm proposed in this paper. A stability analysis and stable policy guidelines are provided. The performance evaluation of BGP-XM, running over an Internet-like topology, shows that high path diversity can be achieved even for limited deployments of the multipath mechanism. Further results for large-scale deployments reveal that the extension is suitable for large deployment since it shows a low impact in the AS path length and in the routing table size.

Keywords: BGP, multipath, interdomain, routing, protocol

1. Introduction

The Internet topology is inherently redundant. Intra-site topologies show high redundancy in their interconnection [1, 2]. Autonomous Systems (ASes) are usually connected to multiple provider ASes [3] to leverage fault tolerance. Internet's increasingly richer connection degree is the result of the quest for improved performance and lower transit costs. Even after removing all the paths which are not usable according to the business relationships established among the connected ASes, a large number of paths traversing different sequences of ASes exists between most of the Internet sites [4]. Additional redundancy comes from the use of multiple links between pairs of neighboring ASes [5].

For years, the networking community has sought for flexible and simple ways to use the largest number of available paths in order to improve availability and perform traffic engineering. In the

Email addresses: jcamacho@it.uc3m.es (Jose M. Camacho), alberto@it.uc3m.es (Alberto García-Martínez), marcelo@it.uc3m.es (Marcelo Bagnulo), fvalera@it.uc3m.es (Francisco Valera)

same way as technologies such as MPLS (MultiProtocol Label Switching) achieve that flexibility in the intra-domain scope, the management of traffic exchanged between domains could be further improved if inter-domain routing policies are changed dynamically.

As reported in [6], this occurs in today's Internet for the particular case of egress traffic in stub ASes. Another example are the mechanisms to coordinate traffic engineering across multiple links between two peering ASes proposed in [7, 8]. These solutions share in common that there is no need to advertise to other domains the routing changes due to the traffic engineering.

Nevertheless, transit ASes must advertise their selected paths to other ASes using BGP (Border Gateway Protocol). In the extreme case, a change in a route selected by a transit AS, which wishes to modify its outgoing traffic pattern, may result in undesired changes for its incoming traffic, because of subsequent routing decisions made by other ASes [9]. Moreover, even if the effects for the incoming traffic are negligible, the deployment of this strategy is likely to stress BGP routers all over the Internet, since they will have to cope with frequent routing changes. In addition to the harm caused by route recomputation, more undesired interactions may appear if ASes perform active path monitoring as in *route flap damping* [10].

Multipath routing has been proposed to achieve flexible and frequent load balancing without path recomputation [11, 12]. Should the routing protocol make available multiple paths to a destination, a router is able to balance traffic across any combination of available paths without being forced to send additional routing messages. This should provide transit ASes with finer control over their outgoing traffic without incurring in the aforementioned problems.

Nevertheless, upgrading the current inter-domain routing to support multipath is far from trivial, especially due to the need of incremental deployments. Current proposals for multipath inter-domain routing are based on the advertisement of additional paths, others than the BGP best path, by means of a parallel negotiation between routers [13] or new BGP capabilities [14]. Unfortunately, their deployment requires the support of the mechanism in two or more neighboring ASes.

Other proposals [15, 16] advocate the utilization of multiple paths while maintaining the BGP advertising scheme, thus announcing only one of them as in-use. Yet, withholding the advertisement of in-use paths requires additional mechanisms to ensure loop-freeness.

The taxonomy is completed with some commercial router implementations, which already allow the use of multiple paths as long as all of them share most of their BGP attributes with the BGP best path [17]. In particular, all the routes to a destination can only differ in their IGP NEXT_HOP attribute. The obtained path diversity mostly result from the use of different links between consecutive ASes.

We claim that a multipath routing mechanism enabling inter-domain traffic engineering must fulfill the following requirements:

Allow high path diversity. The mechanism should impose as few restrictions as possible to the selection of multiple routes. In particular, it must not be restricted to select paths with the same sequence of traversed ASes or paths received from the same neighboring AS, since transit ASes may require moving traffic freely from one neighboring AS to another, e.g., to offload traffic to another provider or to obtain better performance.

Be backwards compatible. The mechanism must allow the selection of multiple routes being advertised by (unipath) BGP routers. Additionally, current BGP routers must be able to receive and use any set of routes created by a multipath router. To fulfill the latter, any set of routes selected by the multipath router must be expressed in a BGP-compatible format. The mechanism must not require any data plane modification in devices other than the multipath-upgraded routers. Hosts and routers must be able to exploit multipath routing without including multipath-

specific information in data packets. Therefore, the resulting multipath service can be described as a multipath *next-hop routing*, i.e., every router takes forwarding decisions independently of the decisions taken by other routers.

Be incrementally deployable. In order to start providing multiple routes, the mechanism must not require being simultaneously deployed in different ASes, or in all the routers of an AS. Enabling multipath operation in a single router should be enough to disclose the multiple paths available to the router.

Be highly configurable. The selection of multiple routes must be tunable. For example, it must allow controlling the size of routing tables, the number of alternative paths per prefix or the prefixes for which multipath routing is enabled. In addition, the multipath mechanism must be configurable to limit the *route stretch*, i.e., the length difference between the longest and the shortest path to a destination, measured in the number of traversed ASes.

Seamlessly preserve usual business relationships. The current business model for inter-domain connectivity results in some widely used relationships such as *transit*, *peering*, *siblings* or *partial transit* [18]. The relationships define preference, import and export filtering rules for route advertisements. Despite the increment in the number of routes, the effort put in the configuration of a multipath mechanism should be close to that for configuring BGP.

Preserve the effects of usual traffic engineering techniques. An AS administrator may choose from a set of techniques to determine how its traffic exits from the AS, and to influence the path followed by the ingress traffic. Domain's inclinations for outbound traffic are usually controlled by associating explicit preferences to routes. The most popular tools to influence the path for the incoming traffic, besides the injection of more specific prefixes, are the use of pre-agreed COMMUNITY values to inform other ASes about local preferences, the artificial increase of the length of the AS_PATH attribute to make the path less attractive, and the use of inter-AS metrics (MED attribute) [19]. Any multipath solution must allow both the domain deploying multipath and the rest of the domains to continue using these tools in a similar way as they currently do.

Generate loop-free paths. Resulting routes must be loop-free. Note that, in order to be backwards compatible, data packets are not required to carry any path selector, so loop-freeness is only assured if none of the possible combinations of the routes selected independently by different routers generate a cycle.

Stable under non-conflicting routing policies. ASes can choose their policies on their own, so the absence of conflicting policies in inter-domain routing cannot be guaranteed [20]. It has been shown that BGP is stable when transit and peering, the most common business relationships, are the only ones deployed [21]. Any multipath routing mechanism must assure that, at least, it is stable in the scenarios described in [21]. Note that this is not a trivial statement, as the work of Chau and Griffin [22] states that multipath can be unstable for configurations in which unipath is stable.

In this paper we present *BGP eXtended Multipath*, BGP-XM, a BGP extension that allows routers to use multiple paths across different ASes, including paths with different next ASes. BGP-XM defines a router architecture tailored to accommodate the information of multiple paths to the same network prefix into a single BGP update message. BGP-XM ensures that the resulting *multipath* updates preserve sufficient and meaningful information, so as to guarantee the routing policy intended by administrators is implemented. Regular tasks, such as policy-based route selection and loop detection are not altered significantly.

In particular, we argue that the benefits of selecting routes received from different ASes and advertising them using an *aggregated* attribute values such as AS_PATH attributes with standard

AS_SET segment type [23], largely compensate for the drawbacks associated with the loss of detailed path description.

We base the design of BGP-XM on the analysis of the specific semantic requirements for intra-site and inter-site dissemination of inter-domain routing information, i.e. IBGP and EBGP modes of operation. From this analysis, we identify under which conditions, relaxing the tie-breaks of the BGP route selection increases path diversity without compromising the AS routing policy.

In addition, the modular BGP-XM architecture allows the definition of different multipath route selection procedures, which may suit different needs. We present K-BESTRO (*K-Best Route Optimizer*) as a possible implementation of the path selection procedure necessary in the BGP-XM architecture.

With these elements, the BGP-XM process architecture fulfills the requirements for an inter-domain multipath routing protocol that we stated above in this section. First, it allows selecting multiple paths which traverse different ASes, increasing the path diversity. It is backwards compatible with unipath BGP routers and enables router-level and AS-level incremental deployments. Moreover, loop detection is possible by inspecting the list of ASes in the generated AS_PATH. Since the semantic of the attributes is preserved, business relationships can still be configured by means of LOCAL_PREF and when combined with appropriate route selection mechanisms (such as K-BESTRO), most standard traffic engineering tools are supported. Also, stability is guaranteed under the conditions stated by Gao and Rexford [21] over BGP policies.

The paper is structured as follows: Section 2 analyzes how different routes can be merged into a single BGP advertisement with the lowest possible impact on the processing of incoming routes, route selection and processing of outgoing routes. To do this, we exhaustively discuss the effect of aggregating the most relevant BGP attributes from multiple routes. Section 3 presents the BGP-XM router architecture, which describes how a BGP-compatible update is generated, and the K-BESTRO route selection mechanism. Then, we provide an example to illustrate how route aggregation is performed by BGP-XM. In section 5, we study in depth the stability properties of our inter-domain multipath framework, and we state one guideline for its safety, which fits with many configurations in use for unipath BGP. After this, the next section, section 6, is devoted to present performance results for BGP-XM for an Internet-like topology, which show that high path diversity can be achieved with just a low number of ASes deploying this architecture. Related work is referenced in section 7, and finally we present the conclusions.

2. Merging multiple paths into a single BGP update

In order to assure compatibility with BGP routers and to allow incremental deployment of inter-domain multipath, the BGP-XM process architecture is designed to generate BGP-compatible update messages.

The challenge in the design arises from the fixed BGP update scheme, which was defined to advertise one path per prefix. Although a BGP router may advertise multiple paths for a particular prefix, some of them even in the same BGP message, only the most recent is considered and it overrides any update previously announced over the same BGP session. As a consequence, the information of multiple paths must be *compacted* into a single update, so as to be advertised.

Specifically, an update is formed by a set of attributes that defines the characteristics of the route. For the majority of these attributes, an update can contain one value at most. If multiple paths are considered, a particular attribute is likely to take different values in each path. Therefore, advertising a multipath set requires a mapping between the attribute value in each path

and a single attribute value. Furthermore, for this operation not to have unwanted effects in the AS routing, the chosen attribute value must be representative of the omitted information in the resulting update.

The most straightforward approach to implement this design is to select only those paths with the same values for LOCAL_PREF, ORIGIN, AS_PATH, and MED attributes. In this case, the attributes of the resulting *multipath* update contain the common values and the NEXT_HOP attribute is valued with the IP address of the multipath router. This may occur (see Fig. 1) if a router (like R2 in AS1) receives routes from different routers belonging to the same neighboring AS (R4 and R5 in AS2), or receiving routes from several routers of its own AS (see R1 in AS1 advertised by R2 and R3), which in turn received their routes from the same AS (AS2 in this case). However, the path diversity disclosed by this mechanism is very limited.

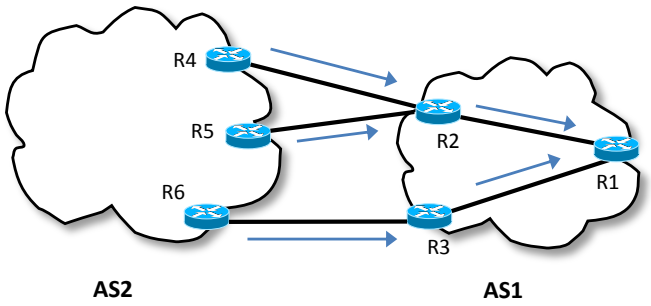


Figure 1: Example of the multipath selection process described in [17]. R2 can select EBGP routes received from R4 and R5, and R1 can select routes IBGP routes received from R2 and R3.

In order to increase the path diversity, we should consider whether it is possible to use paths with different values of LOCAL_PREF, ORIGIN, AS_PATH and MED. Provided that, each attribute plays an intended role in the routing policy, it is essential to understand how this routing information can be merged without breaking their semantic or causing undesired routing configurations.

In the rest of this section, we discuss the implications of selecting routes with different values for the same attribute, and how they can be aggregated into a single BGP update. We analyze the constraints of attribute merging, i.e. when a router must avoid selecting multiple paths, mainly with regard to the preservation of current business relationships and traffic engineering configurations. Since the BGP routing behavior mainly comes from the order of the path selection rules, we follow the same order to present how attributes from a set of paths could be included in a single advertisement. A discussion about the COMMUNITY attribute, not explicitly included in the route selection rules, is also included at the end of the section.

Selection of multiple routes with different LOCAL_PREF values. LOCAL_PREF is used to express the preferences of the network manager. The manager should set the same LOCAL_PREF value for routes which could be concurrently selected in case multipath is deployed. Therefore, multiple routes with different LOCAL_PREF values must not be selected.

Selection of multiple routes with different AS_PATH. Being able to use routes with different AS_PATH should increase the path diversity, since paths discarded before for not meeting the

shortest AS_PATH length criteria are now considered to forward traffic. Although using non-shortest paths may seem inefficient, studies show that in inter-domain routing, a difference of few ASes (e.g. one or two) in the AS_PATH length does not affect the end-to-end performance significantly [9]. BGP-XM should be able to select routes with the same AS_PATH, different AS_PATH of the same AS_PATH length, and even different AS_PATH with different length.

The AS_PATH attribute is used for several purposes: loop detection, inbound filtering, load-balancing, traffic engineering, and it may also be used to indentify the AS number of the neighboring AS to enforce business relationships and MED comparison. Routes with different AS_PATH could be aggregated as long as the previous functions are not impacted. Under the following conditions, multipath does not affect those functions.

In order to avoid loops when multiple routes with different AS_PATH are selected, every AS number included in them should be propagated to other ASes. Including all the AS numbers of every path into a sequence of AS (used to indicate that the route traverses all those ASes and in that precise order) has the undesired effect of making the AS_PATH longer, therefore less attractive to other routers selecting paths according to the AS_PATH length rule.

An alternative to encompass every intermediate AS number into the multipath AS_PATH attribute is to use an AS_SET structure. AS_SETs preserve every AS number, although they neither preserve their order nor indicate that everyone is traversed by the traffic. Note that the latter structure is used by the *prefix aggregation process* [23], which allows combining the characteristics of routes for different prefixes in a way that a single route can be advertised. Although this mechanism was intended to generate a single route from the routes of adjacent prefixes, there is no impediment to use it to aggregate different routes for the same prefix.

In order to preserve the load-balancing and traffic engineering features provided by means of the length of the AS_PATH attribute, we suggest the following guidelines:

- The AS_PATH length of the aggregated route must be the same to that of the available shortest path. In this way, routers applying the AS_PATH length selection rule equally prefer aggregated and non-aggregated routes.
- In the selection, a multipath router must rank higher routes with shorter AS_PATH length over routes with longer AS_PATH length. In this way, AS_PATH prepending could still be used to make some routes less attractive than others.
- In the aggregation, the difference in the length allowed between routes being aggregated should be configurable, otherwise paths may get arbitrarily long. To do so, we introduce later the *Unequal Length MultiPath* parameter (see Section 3.2).
- To conclude, we note that when two or more paths are aggregated, the information about the next AS in the path, which is usually the leftmost AS number in the AS_PATH, is lost (except for one of them or unless it is the same for every path). The normal reported practice to process the neighboring AS information is the use of the COMMUNITY attribute, which is transported across the AS [24, 19]. We encourage the use of this attribute to preserve routing information.

Selection of multiple routes with different ORIGIN. There is no fundamental impediment to mix routes with different ORIGIN values, since this traffic engineering tool for influencing incoming traffic can be easily replaced by other means. When routes with different ORIGIN values are selected, the ORIGIN value of the resulting advertisement can be generated according to the aggregation process [23], i.e., equal to the highest value amongst the selected routes.

Selection of multiple routes with different MED. MED is used by an AS to express the preference for incoming traffic over the different routes which it advertises to a neighboring AS. Whether an AS accepts or ignores MED values from its neighbors depends on their business relationships. For example, some ASes disable MEDs to ensure *hot-potato* routing, i.e., closest-exit routing, widely performed for egress traffic in settlement-free peering relationships between ASes. MED can also be disabled or modified to avoid unintended interactions with other selection rules or prevent route oscillations [25].

When this attribute is considered, the MED semantic intended by the neighboring AS must be honored. Hence, only routes from an AS with the lowest (most preferred) MED should be selected to course traffic. In the multipath case, the same semantic applies but we have two different scopes. Dealing with each neighboring AS, if the neighboring AS would like to receive traffic across many links, it should assign the same (lowest) MED to all of them. Regarding all neighboring ASes, only the paths with the lowest MED values from each neighboring AS should be considered to route traffic.

Unfortunately, the selection of multiple routes from different neighboring ASes, even though they have the same MED, results in a serious problem when this information is to be packed into a single BGP advertisement. It is worth noting that the MED attribute is only meaningful when associated with a neighboring AS, since only with this information it can be determined if MED comparison is necessary between two paths. The BGP specification [23] states that the neighboring AS needed for the MED comparison is determined from the AS_PATH and as discussed above, aggregating two or more routes with different AS_PATHs would prevent identifying the AS to which each MED is associated.

We distinguish two situations. First, when paths are coming from EBGP sessions, the router can identify the neighboring AS from the leftmost AS number in the AS_PATH attribute. The problem comes after the aggregation. When EBGP routes carrying MED are selected, we must ensure that it is possible to identify (the AS number of) the neighboring AS using MEDs and the rest of the routers of the AS can identify it, as well. The reason is that IBGP speakers inside the AS may have to carry out MED comparison. As mentinoed above, since the neighboring AS number cannot be always preserved, multiple routes received through EBGP can be selected if and only if they come from the same neighboring AS and have the same MED.

Second, a router receiving multiple paths through one or more IBGP sessions (and assuming the EBGP aggregation is valid) does not need to apply the same restriction. IBGP information is only forwarded over EBGP sessions and the local AS is appended to the AS_PATH. MED values are dropped, as they are not meaningful to non-neighboring ASes and, if defined, replaced by local MED values. The reader may wonder about the case in which EBGP and IBGP routes are mixed, carrying MED information, but is referred to the next item to see that mixing EBGP and IBGP routes is not allowed in BGP-XM. Note that the resulting behavior is analogous to the behavior of unipath BGP, and so, it is also vulnerable to MED oscillation [25].

Fig. 2 shows an example MED aggregation. R3 can select either the route received from AS2 or the two routes received from AS3 (because they have the same MED), but not all of them. In the figure, R3 selects the routes received from AS3, and generates a single advertisement with MED equal to 0, and AS3 as leftmost to the AS_PATH, so that the AS number associated with the MED can be identified. Those paths are redistributed to R1 by means of IBGP. As pointed out before, routing information learnt from IBGP sessions is not redistributed to other IBGP speakers and it is only propagated further over EBGP sessions. For that reason, the potential inconsistency in the MED comparison cannot take place in other routers in the AS. In the example, R1 can select both routes received from R2 and R3, even though they have different MED and they have

been received from different neighboring AS. Outside the AS, the AS_PATH is extended with the local AS, becoming the new next AS in the path.

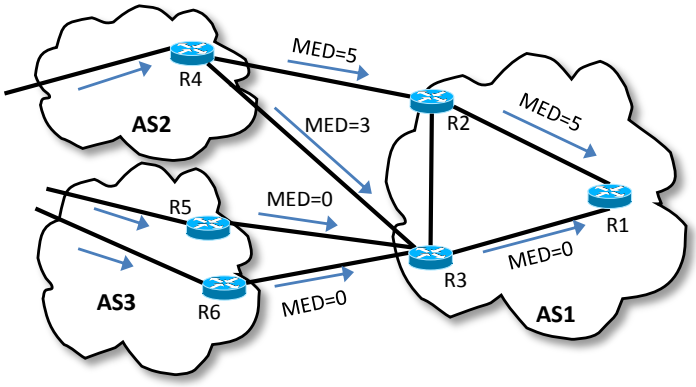


Figure 2: Example of aggregation of MED information for a given prefix inside an AS with BGP-XM.

A less restrictive scenario for aggregation occurs when MED comparison is not performed because MEDs are set to zero, i.e. not defined. In this case, a configuration knob could be set to allow any path to be aggregated, regardless the AS from which it has been received. This configuration is also much safer in terms of oscillations, although may not fulfill certain traffic engineering requirements.

Selection of multiple external/internal routes. The BGP route decision process is configured to prefer EBGP over IBGP routes, and thus perform hot-potato routing. Although we could think of selecting both EBGP routes and IBGP routes in order to increase path diversity, mixing IBGP with EBGP paths allows two multipath routers to select each other as egress points of the AS, thus creating a forwarding loop.

Unless additional mechanism, e.g. MPLS tunneling is available, either EBGP-only or IBGP-only routes can be selected.

Selection of multiple routes with both different NEXT_HOP and different costs to the NEXT_HOP. According to the external/internal route selection analysis, we can distinguish two cases: selecting among different routes received from EBGP (in which case any IBGP route would have been removed), and selecting among different routes received from IBGP. Note that, in the considered multipath model, a router only propagates one advertisement (possibly, a multipath advertisement) to a neighboring router. Therefore, a single NEXT_HOP is associated with the multiple routes selected by a router, and router does not receive different advertisements with the same NEXT_HOP.

For the EBGP case, the network manager is responsible for configuring the cost metric of the inter-domain links. In case he wanted to allow the selection of multiple routes coming from different EBGP routers, he can configure the same link metric to these external routers. If he does not want to aggregate routes coming from different EBGP routers, he can set different metrics for the links connecting to them.

For the case in which multiple IBGP routes are considered, we could initially expect that the manager may want to aggregate routes received from different IBGP neighbors although they may have different costs to the NEXT_HOP, to increase path diversity. However, such behavior may result in internal forwarding loops. We illustrate this issue with the topology depicted in

Fig. 3. Each router executes the multipath BGP route selection process, and routers R1 and R2 use the minimum cost to the NEXT_HOP to select the path or paths to the destination prefix. If routers are allowed to select multiple routes with different costs to the NEXT_HOP, R1 and R2 could both select R3 and R4 as egress routers, provided that the cost of the R1-R2 link is low enough. In this situation, a forwarding loop is created between R1 and R2, since each router can send traffic to the other to arrive to the same destination prefix.

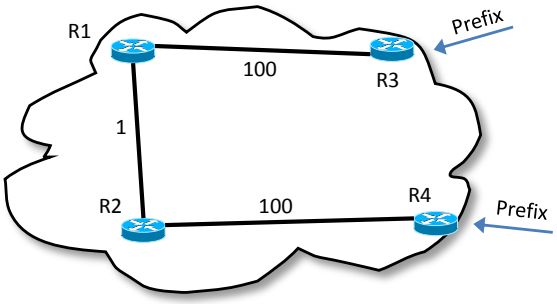


Figure 3: Example of internal AS topology.

It is well known that a sufficient condition for loop-freeness inside the AS is to select paths with the best cost to the NEXT_HOP addresses of the different routes. To ensure that route aggregation occurs for the desired cases, internal costs can be manipulated accordingly, as it is usually done with ECMP routing [26][1].

A particular case for the selection of multiple routes received through IBGP occurs when the internal AS connectivity relies on end-to-end tunnels (e.g., MPLS) between border routers. Loops are not possible, so every route received from IBGP could be selected.

Finally, both for EBGP and IBGP, the router performing the aggregation generates an advertisement with one of its own IP addresses as NEXT_HOP.

Selection of multiple routes with different NEXT_HOP (coming from different routers, links, etc.) and same cost to them. BGP tie-breaks assure route uniqueness by discarding routes according to BGP identifiers, link addresses, etc. All these routes can be aggregated by a multipath inter-domain routing mechanism. None of these identifiers for the received routes need to be propagated further.

Selection of multiple routes with different COMMUNITY. The COMMUNITY attribute can be used when performing ingress filtering, attribute modification prior to route selection and egress filtering, although it is not explicitly involved in the route selection process. The aggregation of routes with different COMMUNITY depends on the semantic of the values, which widely differ. We assume that routers do not propagate COMMUNITY which are not understood,

[1]However, note that the inter-domain multipath routing deployment is fundamentally different from ECMP, since in the first case routes are built to different egress points. The combination of ECMP and inter-domain multipath routing in the site may provide additional paths, since multiple internal paths can be used to reach to a single NEXT_HOP, for which a single route has been propagated by multipath BGP.

as it is recommended by [24], with the COMMUNITY being removed prior to any other route processing.

For COMMUNITY values that refer to LOCAL_PREF or MED settings for the current AS, the COMMUNITY does not need to be processed anyway, since the route is selected or not according to the induced LOCAL_PREF or MED values. After being mapped into their corresponding LOCAL_PREF or MED values, these COMMUNITY values can be removed. For COMMUNITY values which refer to LOCAL_PREF or MED settings for other ASes, values which are allowed to traverse the local AS, only routes with equal COMMUNITY values should be allowed.

We now analyze the case of the transport and selection of COMMUNITY values indicating the type of relationship maintained with the AS from which the route is received. In this case, the COMMUNITY indicates whether the route was received from a customer or from a peer/provider. Such information is local to the AS, and it is generated by a router, which has selected one or many EBGP routes. According to usual relationships, routes received from neighboring ASes playing different roles must be associated with different LOCAL_PREF values in the router, so they should not be aggregated. Hence, an advertisement will only be associated with one of these values. For the router receiving multiple advertisements, again different LOCAL_PREF values will be associated with different COMMUNITY values, so just regular route selection will result in selecting routes from the same type.

For COMMUNITY values used to convey filtering information (e.g., NO_ PEER, or per-AS filtering indications) to perform traffic engineering for ingress traffic, only routes with the same filtering policy should be aggregated. The same occurs for COMMUNITY values indicating geographic location or the Internet eXchange Point (IXP) from which the route was received [27]. The network manager could analyze in a case-by-case basis if he could filter out or merge some of these values in order to increase path diversity.

Summary. We now state some conclusions regarding the selection and aggregation of multiple routes. In order to respect the intentions expressed by the network manager, we should not aggregate routes with different LOCAL_PREF, different MED for routes received from the same neighboring AS, or routes with different values for some semantic associated with the COMMUNITY. Due to the problems associated with MED semantic and attribute aggregation, we recommend aggregating routes received through EBGP and containing MED only when they have the same MED and come from the same neighboring ASes. To assure loop-freeness inside a site, we recommend that either EBGP-only or IBGP-only routes should be aggregated, and among them, aggregate only routes with equal cost to the NEXT_HOP, unless loop-freeness is provided otherwise (e.g., by use of MPLS).

Therefore, path diversity may result from the aggregation of routes with different ORIGIN and AS_PATH values, selecting multiple routes with the same MED coming from the same AS, selecting multiple EBGP (IBGP) routes, or routes with the same distance to the NEXT_HOP. Note that two routes differing at the same time in their ORIGIN, AS_PATH values and distance to the NEXT_HOP, for example, could be selected. In order to benefit from the aggregation of routes with different AS_PATH, the aggregation mechanisms defined in BGP can be used. No modification in the format used to exchange information is required for our solution.

3. BGP-XM architecture

As discussed before, there are a number of cases in which different routes can be selected at the same time without any impact in the operation of other BGP routers. BGP-XM defines

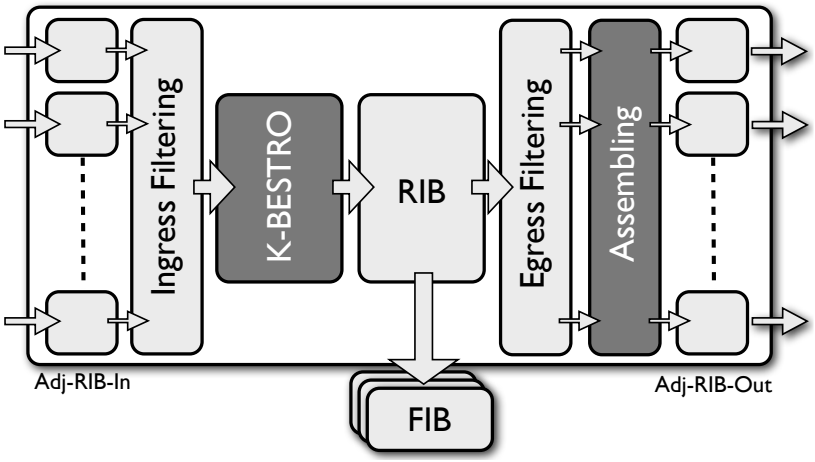


Figure 4: BGP-XM router architecture

an architecture that extends the BGP routing process to allow the use of these multiple routes. The design reuses most of the BGP protocol, e.g., session management, filtering and messages format. Fig. 4 shows the module diagram of the BGP-XM router architecture.

As for BGP, candidate paths for a prefix are retrieved from the *Adj-RIB-In* modules and undergo the *Ingress Filtering*.

Afterwards, the path selection process is executed and the resulting multipath set is installed in the RIB (Routing Information Base) of the router. The RIB completes the routing information from other routing processes, e.g., the IGP, and the FIB (Forwarding Information Base) updates the entry for the prefix. Ideally, as long as the semantic of BGP attributes is preserved in the selection process and in the route advertisement which represents the multipath set, BGP-XM should work with any arbitrary multipath decision process, following the approach in [28]. In this paper we propose one particular decision process, which we term *K-BEST Route Optimizer* (K-BESTRO). This selection process provides up to K best paths which can differ among them in their ORIGIN, AS_PATH length and next AS, allows the selection of multiple routes being either EBGP or IBGP. Also, they can be part of the multipath set routes with the same distance to the NEXT_HOP. K-BESTRO, as any multipath decision process to fit in the BGP-XM architecture, must select routes, which could be later assembled into a single BGP update.

For each ongoing session, egress filters are defined like in BGP. We have reasoned in our previous analysis (see Section 2) that the route selection process should only select routes that would be propagated in the same conditions. This is a direct consequence of the way business relationships are reflected in the routing configuration, i.e., each type of neighboring AS (e.g. customer, peers, providers) is subject to the same export policies. Hence, it is possible that for some egress filtering configurations, BGP-XM filters must discard the whole multipath set as soon as one path is not compliant with the export policy.

Prior to advertising, the new module *Assembler* performs the merging of the multipath information into a BGP Update message. That *assembling* operation allows regular BGP routers to process BGP-XM Updates and avoids AS path loops.

We next describe the Assembler and the K-BESTRO modules.

3.1. Route dissemination: The assembler module

This section describes how BGP-XM advertises multiple paths using BGP Update messages. The design of Assembler is aligned with the BGP prefix aggregation [23] philosophy. As mentioned in the previous section, the operation of Assembler enables BGP-XM routers to select paths whose attributes have different attribute values (and still can be aggregated). It merges the multipath routing information so that the values included in the resulting BGP Update are the aggregated values of those in the individual paths.

Since the aggregation is already part of the BGP standard as described in [23], backwards compatibility is guaranteed. BGP routers can parse the announced BGP messages and update their routing information (e.g., to assure loop detection) and in the meantime, multipath-capable routers can concurrently use different paths towards the same prefix. In addition, Assembler inherits the scalability properties of the prefix aggregation mechanism, i.e., regardless the amount of paths used by a router, its peering routers install only a single entry in their routing tables. The latter avoids the exponential growth of the routing table entries.

The Assembling algorithm performs the aggregation of some attributes as defined in [23]. It sets the NEXT_HOP attribute to one of the addresses of the router running the algorithm. It also sets the ORIGIN to the maximum value of this attribute for all the routes to aggregate. LOCAL_PREF and MED must be the same for all selected routes, so the corresponding value is included in the new advertisement. Regarding the AS_PATH attribute, different algorithms can be used for its aggregation [23]. For BGP-XM we suggest the *Path Assembling* algorithm shown in Table 1.

The algorithm starts selecting a path with the shortest AS_PATH length, SP, as described at rule 1 in Table 1. Then, a new empty AS_SET is created. Going through every path being assembled, AS numbers that are not already in the shortest path are added to the new AS_SET. If the shortest path already contained an AS_SET, then both AS_SETs are merged. Otherwise, the newly composed AS_SET is appended to the back (i.e. rightmost part of) the shortest path. Once all the paths have been processed, if the AS_SET is not empty, the AS_PATH is one AS number longer than the original shortest path SP. To solve this, the rightmost ASN in the AS_SEQUENCE of the shortest path SP is moved to the AS_SET.

Lets consider the following examples: When aggregating $P_1 = (1, 2, 3, 8)$, $P_2 = (4, 5, 6, 8)$, $P_3 = (7, 8)$, the shortest path P_3 does not contain an AS_SET. The algorithm aggregates P_1 and P_2 into the AS_SET $\{1, 2, 3, 4, 5, 6\}$ and appends it to P_3 to create the assembled path $P' = (7, \{8, 1, 2, 3, 4, 5, 6\})$. Notice that ASN 8 is moved into the set to keep the length equal to 2. Another example is the aggregation of $P_1 = (1, \{2, 3\})$, $P_2 = (4, 5, 3)$, $P_3 = (6, 7, 3)$. The shortest path in this case contains an AS_SET. The resulting AS_PATH is $P' = (1, \{2, 3, 4, 5, 6, 7\})$.

The generated AS_PATH has the same length as the shortest AS_PATH within the multipath set to avoid the penalty that BGP applies to longer paths. Also, the path structure follows the most observed pattern in the Internet for aggregated paths [3, 29], which is an AS_SEQUENCE followed by an AS_SET.

Regarding to the COMMUNITY values of the assembled advertisement, many cases could occur depending on the semantics of the received communities and on the configuration of the router: new values may be generated, all the routes may have the same COMMUNITY, etc. In any case, a specific configuration should be required to define this behavior, according to the guidelines presented in section 2.

Table 1: Assembling Algorithm

1.-	Pick up one of the shortest paths from the multipath set. Lets this path be SP.
2.-	Create an empty *AS_SET* $\mathcal{S}$.
3.-	For every ASN from other paths and not present in SP, add it to *AS_SET* $\mathcal{S}$.
4.-	If SP already contained an *AS_SET*, merge it with $\mathcal{S}$.
5.-	Else, if $\mathcal{S}$ is not empty, move the rightmost AS to $\mathcal{S}$ and append $\mathcal{S}$ to SP.
6.-	If the assembled path is advertised through an EBGP sessions, prepend the local AS number to the *AS_PATH*.

3.2. Path selection process: The K-BESTRO module

The path selection process is, together with the Assembler, the multipath enabler element of BGP-XM. In this subsection, we present our particular design of the selection process, called *K-Best Route Optimizer*, K-BESTRO. K-BESTRO operates in a four-phase process: First, it discards routes which are not acceptable from a routing policy perspective, either defined explicitly by network managers by means of LOCAL_PREF and MED, removing routes with a variation in AS_PATH length exceeding a (relative) maximum value defined by the *Unequal Length MultiPath* parameter (i.e., ULMP), or removing IBGP routes when EBGP routes exist. In the second phase, it ranks the remaining routes according to some criteria. Then, it incrementally builds a set of routes which can be assembled together in a single BGP update, discarding the routes that do not fit into that set. Finally, it takes the first K routes remaining from the previous process, in order to limit the amount of resources consumed by multipath routing operation. The rules applied are described in Table 2. We next discuss in detail each phase:

3.2.1. Policy filtering rules

The *policy filtering rules* are responsible for discarding the routes that do not fulfil the specified routing policy. In addition to the routing policy specified by means of LOCAL_PREF and MED, a novel *multipath policy* is added to the design to let administrators define how much variability they want to allow in the maximum AS_PATH length of the accepted routes.

The algorithm keeps the paths assigned by the administrator with the highest local preference. After that, the AS_PATH length is evaluated. The algorithm allows the selected paths to deviate from the shortest path behaviour. As suggested in [9], a difference of few ASes in the AS_PATH does not imply a worse end-to-end path quality in practice. Moreover, by limiting the maximum length deviation by means of the ULMP parameter, traffic engineering using path prepending can be supported (although more prepending is needed).

As for BGP, the MED comparison is performed afterwards. For each neighbor AS, the paths with the lowest MED are selected.

Rule 4 states that the algorithm always prefers EBGP over IBGP paths, since the larger the amount of EBGP paths the better the diversity of the multipath set should be. Finally, rule 5 prefers routes with lowest distance to the NEXT_HOP.

The resulting set of paths undergoes after the set of ranking rules.

3.2.2. Ranking rules

The multipath policy also specifies the maximum number of paths that can be chosen for a prefix. The number of paths selected at this stage of the decision process may exceed that limit and extra paths must be removed. In order to obtain a predictable outcome from this second stage

of the path selection, additional criteria are defined to apply a (lexicographical) order over the paths, creating a ranking.

Rule 6 in Table 2 depicts the ranking rules. They are based on the criteria behind the BGP tie-breaking rules, however the difference stems from the creation of an merit-based order rather than discarding paths. Ties in the ranking positions are solved by evaluating consecutively the following attributes: A higher rank is given to paths with, shorter AS_PATH length, then lower ORIGIN, lower BGP identifier and finally lower network address.

3.2.3. Assembling filtering rules

The Assembler module merges the information of aggregatable attributes from different paths, so that the multipath set chosen by K-BESTRO can be more diverse. Nonetheless, as pointed out in section 2 for some other attributes, such as the MED and some BGP COMMUNITY values, the paths selected by K-BESTRO must have the same value in order to be assembled. The Assembling Filter enforces the necessary conditions over the ranked paths, such that they can be assembled. That implies that in some cases routes must be discarded.

The attribute values of the *highest ranked path*, i.e., HRP, are taken as reference for the aggregation. The Assembling Filtering rules ensures that after the filtering, all remaining paths in the ranking can be aggregated with the HRP. If one attribute is not present in the HRP, then only paths without that attribute remain in the rank. The MED should be always in the Assembling Policy (unless the AS does not honor the MED of its clients). Rules 7.a. and 7.b. in Table 2 define the filter for the MED. Rules 8.a. and 8.b. show an example over the NO_EXPORT community. Additional rules can be added for other COMMUNITY values.

3.2.4. K-Best selection rule

In order to select the best possible multipath set, this sub-module runs through the rank and (if available) selects up to K paths, where different K values can be defined for each prefix in the multipath policy.

4. Example: An AS running BGP-XM

This example is referred to Fig.5. The figure represents a transit AS (AS1) with three customers (AS3, AS4, AS6) and one provider (AS2). AS1 BGP-XM border routers are configured to comply with the filtering and preference rules usually defined to deploy business relationships [21]. All the routers at AS1 have ULMP=1 (i.e., they select only paths with shortest AS_PATH length, and shortest AS_PATH length plus 1) and KBEST=3 (i.e., they select up to three routes for a given destination). Routers establish a full-mesh of IBGP sessions to redistribute routing information. The IGP distance is the same between every IBGP routers in AS1 and for every link connecting to the external ASes. The example presents how the prefix 160.1/16 is propagated from AS5 to AS2. Paths selected by routers are shown in solid arrows, while those not selected, although available, appear in dashed style. Comma-separated numbers represent the value of the AS_PATH attribute.

AS3 advertises three paths directly to AS1: two to R2 with MED=10 and MED=20, and another one to R4 with MED=10. AS4 does not use MED, but prepends twice its own AS number in the update to R9. AS6 propagates its route to R10. Every path in AS1 is assigned the same local preference.

R4 can select EBGP paths across AS3 and AS4. Nevertheless, those paths cannot be assembled since one has MED and the paths go through different ASes (rule 8.a in Table 2). Therefore, the

Table 2: K-BESTRO route selection process

	Policy Filtering Rules
1.-	Keep paths with highest LOCAL_PREF
2.-	Look for the shortest AS_PATH length, define its length as *shortest-l* and keep paths with *length* <= *shortest-l*+ ULMP.
3.-	For each advertising AS, look for paths with lowest MED
4.-	If among the remaining paths there is one with session type EBGP, delete paths with TYPE IBGP
5.-	Keep paths with lowest distance to NEXT_HOP
	Ranking Rules
6.-	Rank the paths according to,
6.a.-	Paths with shortest *AS_PATH* length ranked higher.
6.b.-	Paths with lowest ORIGIN ranked higher
6.c.-	If equal cost, lowest BGP identifier ranked higher
6.d.-	If same BGP identifier, lowest peer address ranked higher
	Assembling Filtering Rules
7.a.-	If FRP has session TYPE EBGP and MED, remove paths from different AS or different MED
7.b.-	Else if FRP has session TYPE EBGP and no MED, remove paths with MED
8.a.-	If first ranked path (FRP) has NO_EXPORT Community, remove paths with different NO_EXPORT Comm.
8.b.-	Else, delete paths with any NO_EXPORT Community
...	*(Additional Checks)...*
	K-Best Selection Rule
9.-	Select K best ranked routes

path through AS4, which has a lower BGP identifier (i.e., R6), becomes the first ranked path for R4. The assembling filter then discards the path through AS3.

Router R2 discards one of its own EBGP paths received from the same AS because of having higher MED. It compares the remaining paths across AS3 with the internal path through R4 and AS4. Since R2 prefers EBGP paths to IBGP, then it discards the IBGP path through R4.

Router R10 applies the route selection process to the routes received directly from AS6 and the other routes received through IBGP. Since ULMP is set to one, none of the routes is discarded according to the AS_PATH length rule. Then, the route received from AS6 is selected according to the rule which prefers EBGP routes over IBGP ones.

Finally, router R9 receives an EBGP path from R3 in AS4, which is discarded because its AS_PATH is longer than 3. R9 selects the IBGP paths from R2, R4 and R10 (it can aggregate up to three routes with ULMP equal to 1), aggregates them and advertises the assembled path through any EBGP session. The aggregated path is created following the algorithm in Table 1. The process is as follows: First, one of the shortest paths is selected, and the last AS number is moved to an AS_SET, e.g., (4, {5}). Then, the AS numbers, not included in the initial shortest path, are added into the AS_SET. Therefore, the resulting AS_PATH is (4, $\{5, 3, 6\}$). Before exporting the path to AS2, R9 appends its own AS number to the AS_PATH.

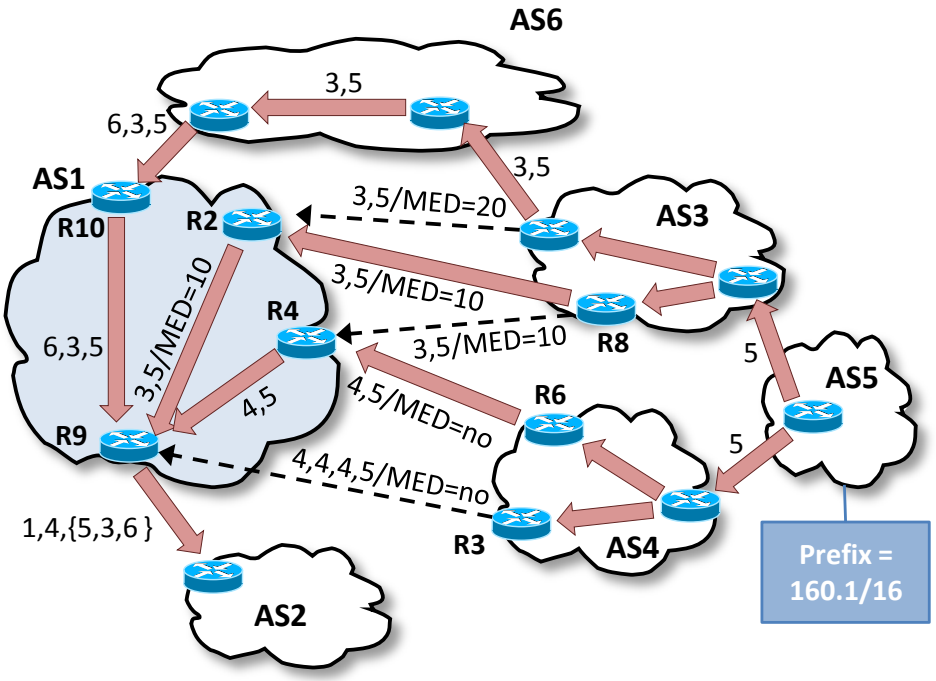


Figure 5: Model of a transit AS with BGP-XM Routers

5. Stability analysis

The stability of BGP is well-known to depend on the preferences of the individual nodes over the available paths. In some cases, the preferences of all nodes cannot be fulfilled simultaneously and nodes may enter into conflicts, so-called *dispute wheels* [20]. Each router chooses a path according to its preferences. When several equally preferred paths are available, a BGP router selects just one of them applying a tie-break.

According to the results in [22], the stability of BGP depends on both preferences and the tie-break rule. In order to select multiple paths and perform multipath routing, the path selection and tie-break rules must be modified. Therefore, the fact that BGP converges to a stable solution does not guarantee that the network is stable if routers select several paths, instead. The relaxation of the selection process may trigger oscillations that did not appear before in the unipath case. This result is the main motivation of this section.

Since BGP-XM selects multiple paths and additional information is advertised between routers. The goal of our stability analysis is to show that the deployment of BGP-XM in a network does not affect the stability. The section starts with a discussion to motivate the stability analysis. Afterwards, the stability of BGP-XM is studied. To do so, we use the framework in [22] and include *assembled* paths to the model. The stability analysis proves that the existing relation amongst AS routing policies must fulfil a property called *anti-reflexivity* so as to assure asynchronous convergence for BGP-XM. In Section 5.3 we use this result to reformulate the guideline proposed in [21], and we re-state a safety condition for BGP-XM that prevents

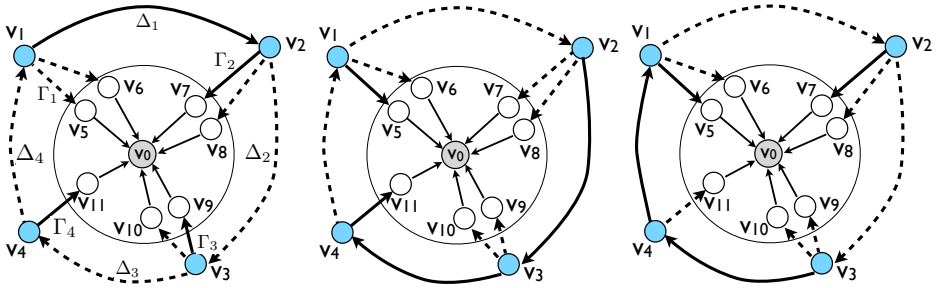


Figure 6: Unipath Dispute Wheel Sequence

	$\overset{\triangleleft}{\approx}$				
v_1:	$v_2v_7...v_0$	$v_2v_8...v_0$	$v_5...v_0$	$v_6...v_0$	$v_2v_3...v_0$
v_2:	$v_3v_9...v_0$	$v_3v_{10}...v_0$	$v_7...v_0$	$v_8...v_0$	$v_3v_4...v_0$
v_3:	$v_4v_{11}...v_0$	$v_9...v_0$	$v_{10}...v_0$	$v_4v_1...v_0$	$v_4v_1...v_0$
v_4:	$v_1v_5...v_0$	$v_1v_6...v_0$	$v_{11}...v_0$	$v_1v_2...v_0$	

Figure 7: Per-Node Preferences that create the Unipath Dispute Wheel

osciallations across different ASes.

5.1. On dispute wheels in unipath and multipath scenarios

The goal of this section is to introduce a discussion about the impact of multipath in the stability of policy-based routing protocols. The notation, adapted from the framework defined in [22], is presented before the discussion.

In policy-based scenarios where a path vector protocol is running, only the most preferred paths propagate from one node (router) to another. When a path P is preferred over a path Q according to the preferences of a node, that relationship of *preference* can be denoted as

$$P \overset{\triangleleft}{\approx} Q \tag{1}$$

If a path $P = v_i, \ldots, v_0$ is announced and it is chosen as most preferred by an arbitrary sequence of nodes $v_{i+n}, \ldots, v_{i+1}$ in the network, the assigned path to v_{i+n} is a propagated version of P, i.e. $P' = v_{i+n}, \ldots, v_{i+1}, v_i, \ldots, v_0 = (v_{i+n}, \ldots, v_{i+1})P$. This relationship of *composition* with P can be expressed as

$$P \overset{\blacktriangleleft}{\approx} P' \tag{2}$$

Using these two simple concepts, different relations among the policies of the different nodes and the paths announced by them can be denoted. A particular type of relations between paths caused by routing policies is the *reflexive* relation defined as follows (for a formal and more rigorous definition of *anti-reflexive* and *reflexive* relations see [22]): If there is an alternating sequence of preference ($\overset{\triangleleft}{\approx}$) and composition ($\overset{\blacktriangleleft}{\approx}$) relations created among a set of paths

$P_1, P_2, \ldots, P_n, Q_1, Q_2, \ldots, Q_n$, it is said that the relation is a *reflexive* relation if there is a path for which the alternating sequence of relations is cyclic, e.g.

$$P_1 \blacktriangleleft\!\!\approx Q_n \triangleleft\!\!\approx P_n \blacktriangleleft\!\!\approx \ldots \triangleleft\!\!\approx \ldots \blacktriangleleft\!\!\approx Q_2 \triangleleft\!\!\approx P_2 \blacktriangleleft\!\!\approx Q_1 \triangleleft\!\!\approx P_1 \tag{3}$$

A significant property of reflexive relations is that all the individual relations cannot be fulfilled simultaneously. For instance the path Q_n composed by an arbitrary path and P_1 is more preferred than P_1, which is a contradiction, since when Q_n is selected P_1 is not, and Q_n is feasible if and only if P_1 is selected. Hence, no (best) solution exists and the protocol may oscillate, as it cannot find the solution for the given configuration.

From [22], if no subset of nodes, preferences and paths create a reflexive relation, then convergence is guaranteed. Since the generation of a dispute wheel involves a specific relation among the ranking functions, by changing the ranking functions and announcing additional paths, the relation among them changes as well, in comparison to the unipath case. A mentioned, a stable-state for a unipath configuration may no longer be reachable when multipath is enabled. Before addressing the formal analysis of the problem, we introduce two examples. The first one shows that the propagation of additional paths can provide a network running BGP-XM with a stable solution even though there is no unipath stable solution. The second example shows a case with stable unipath solution but where the use of multipath routing may activate a dispute wheel.

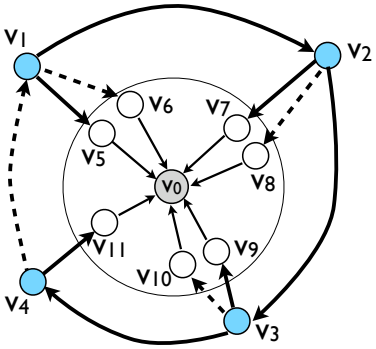
Example 1. In Fig.6, a network is depicted in which every node selects only one path following the same criteria as BGP. The path ranked in first position by each node is displayed with a solid arrow, whereas the rest of feasible paths (with lower-ranked) are displayed in dashed arrows. Nodes inside the circle ($v_5 - v_{11}$) have at least one stable path to the destination node v_0, i.e., the path belongs to the final path assignment. Nodes v_1, v_2, v_3, v_4 are one-hop away to the set of stabilized nodes, and they do not have an stable path assignment yet. The preferences of each node are shown in the table at Fig.7. For each node, the leftmost paths in the row are the most preferred. In Fig.6-left the node v_1 receives three paths towards v_0 through v_2, v_5 and v_6 respectively. The three paths are of the same AS length and v_1 chooses the path $v_2 v_7 \ldots v_0$ after applying the lowest BGP_ID route selection rule. Node v_2 has three paths of equal path length through v_7, v_8 and v_3. It is not aware yet of the path $v_3 v_9 \ldots v_0$ and chooses to go through v_7 using the lowest BGP_ID criteria. Node v_3, configured in a similar way, chooses v_9 as next-hop since it is not aware of the path $v_4 v_{11} \ldots v_0$. Node v_4 receives a path through v_{11} but it is not aware of the path $v_1 v_5 \ldots v_0$, therefore it chooses v_{11} even though its highest preference is to use $v_1 v_5 \ldots v_0$. In addition, v_4 filters any AS_PATH containing v_2.

In Fig.6-middle, v_2 becomes aware of the path through v_3 and changes its assignment, forcing v_1 to change its path as well. Node v_1 does not select the new path of v_2 because it is longer than the selected path through v_5. However, v_3 becomes aware of the path through v_4 and changes as well. The path assignment is again modified in Fig.6-right. Node v_2 looses its path $v_2 v_3 v_9 \ldots v_0$ as it is longer than $v_2 v_7 \ldots v_0$ and v_4 prefers the path announced by v_1. In the next step, the nodes go back to the initial assignment shown in Fig.6-left completing a cycle in the oscillation. The reflexive relation for this unipath configuration can be expressed in this case as follows

$$\begin{aligned} v_1 v_5 \ldots v_0 \blacktriangleleft\!\!\approx v_4 v_1 v_5 \ldots v_0 \triangleleft\!\!\approx v_4 v_{11} \ldots v_0 \blacktriangleleft\!\!\approx \\ \blacktriangleleft\!\!\approx v_3 v_4 v_{11} \ldots v_0 \triangleleft\!\!\approx v_3 v_9 \ldots v_0 \blacktriangleleft\!\!\approx v_2 v_3 v_9 \ldots v_0 \triangleleft\!\!\approx \\ \triangleleft\!\!\approx v_2 v_7 \ldots v_0 \blacktriangleleft\!\!\approx v_1 v_2 v_7 \ldots v_0 \triangleleft\!\!\approx v_1 v_5 \ldots v_0 \end{aligned} \tag{3}$$

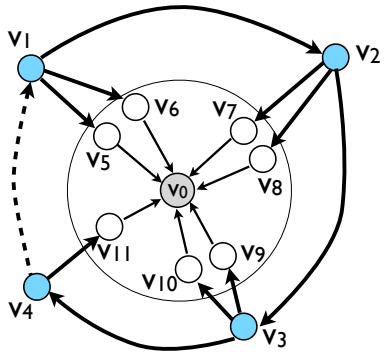
For the same scenario as in Fig.6 and with preferences relaxed to the multipath case, the K-BESTRO selection algorithm is tuned to select 2 paths with the same AS_PATH length. The scenario becomes stable because every node chooses the path through one neighboring node on the stable set, and a path through an non-stabilised neighbor (clockwise) except for v_4. For instance, v_1 selects the paths through v_5 and v_2. Fig.8 shows the final path assignment. Node v_4 selects only the path through v_{11} since the advertisement from v_1 contains v_2, to which v_4 assigns a lower preference (see the preferences table on the right).

Finally, if BGP-XM neither constrains the path length nor the amount of paths, the stable path assignment displayed in Fig.9 can be achieved.



	$\overset{\triangleleft}{\approx}$		
v_1:	$v_2 \ldots v_0$ $v_5 \ldots v_0$	$v_2 \ldots v_0$ $v_6 \ldots v_0$	$v_5 \ldots v_0$ $v_6 \ldots v_0$
v_2:	$v_3 \ldots v_0$ $v_7 \ldots v_0$	$v_3 \ldots v_0$ $v_8 \ldots v_0$	$v_7 \ldots v_0$ $v_8 \ldots v_0$
v_3:	$v_4 \ldots v_0$ $v_9 \ldots v_0$	$v_4 \ldots v_0$ $v_{10} \ldots v_0$	$v_9 \ldots v_0$ $v_{10} \ldots v_0$
v_4:	$v_{11} \ldots v_0$ $v_1 v_5 \ldots v_0$ $v_1 v_6 \ldots v_0$	$v_1 v_2 \ldots v_0$ ...	

Figure 8: Stable solution when nodes select their best 2-paths.



	$\overset{\triangleleft}{\approx}$		
v_1:	$v_2 \ldots v_0$ $v_5 \ldots v_0$ $v_6 \ldots v_0$	$v_5 \ldots v_0$ $v_6 \ldots v_0$	...
v_2:	$v_3 \ldots v_0$ $v_7 \ldots v_0$ $v_8 \ldots v_0$	$v_7 \ldots v_0$ $v_8 \ldots v_0$	...
v_3:	$v_4 \ldots v_0$ $v_9 \ldots v_0$ $v_{10} \ldots v_0$	$v_9 \ldots v_0$ $v_{10} \ldots v_0$	...
v_4:	$v_{11} \ldots v_0$ $v_1 v_5 \ldots v_0$ $v_1 v_6 \ldots v_0$	$v_{11} \ldots v_0$	$v_1 v_2 \ldots v_0$ ...

Figure 9: Stable solution, this time every node uses its maximum number of available paths.

Example 2. In this second example we want to show the oppossite situation: it may occur that there is a unipath solution, but there is no multipath one. Nodes v_1 and v_2 are running BGP-XM to select up to 2 paths with an AS_PATH length difference of at most one. Assume that, v_1 and v_2 give the same preference to any of their connections. The policies in that case can be expressed as in the table on the right side of Fig.10. The reflexive relation is in this case

$$v_1 v_0 \overset{\blacktriangleleft}{\approx} v_2\{v_1, v_0\} \overset{\triangleleft}{\approx} v_2 v_0 \overset{\blacktriangleleft}{\approx} v_1\{v_2, v_0\} \overset{\triangleleft}{\approx} v_1 v_0 \tag{4}$$

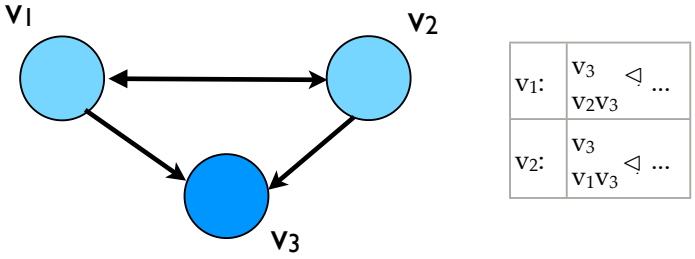


Figure 10: Scenario with unipath but not multipath solution.

The reflexive relation appears due to the fact that both nodes prefer the aggregated of the direct and indirect paths, rather than only the direct path. For an equal-length multipath configuration, a solution exists in which BGP-XM assigns only direct paths to each node.

These two examples lead to the conclusion that the propagation of additional paths can either stabilize a network or activate a reflexive relation. Therefore, it is necessary to analyze the stability of BGP-XM, since the stability results inferred for BGP do not apply.

5.2. Convergence of BGP-XM

Let $G = \langle V, E \rangle$ be a topology graph, where V is the set of vertex, E is the set of edges of the graph, and $v_0 \in V$ denotes the origin of a prefix advertisement. Let $\mathcal{P}(v_i, v_0)$ be the set of *reachable* paths between v_i and v_0 in G, i.e. any path that can be constructed hop-by-hop from v_i to v_0. Now $\mathbb{P}(\mathcal{P}(v_i, v_0))$ is defined as the super-set of all the permutations of the subsets in $\mathcal{P}(v_i, v_0)$, so to speak, every element $\Phi_{v_i} \in \mathbb{P}(\mathcal{P}(v_i, v_0))$ is an arbitrary subset of elements in $\mathcal{P}(v_i, v_0)$. A *partial multipath assignment* is defined as

$$\mathbf{\Phi} = \{\Phi_{v_i} \in \mathbb{P}(\mathcal{P}(v_i, v_0) \cup \emptyset), \forall v_i \in V\} \tag{5}$$

Note that for some nodes the assignment may be empty ($\emptyset$ represents the empty set). The protocol is modelled as a fixed-point iteration of a distributed synchronous Bellman-Ford mapping $\mathcal{F}(\mathbf{\Phi})$ over the multipath assignment $\mathbf{\Phi}$.

The protocol starts growing from the initial iteration, in which the origin v_0 announces the path containing itself to its neighbors, and it increases the path assignment until reaching an assignment $\mathbf{\Phi}$ that verifies the fixed-point equation,

$$\mathbf{\Phi} = \mathcal{F}(\mathbf{\Phi}) \tag{6}$$

In the BGP-XM mapping, nodes advertise assembled paths as depicted in section 3.1. The most recent advertisement of an assembled path received by node v_i from node v_j at iteration k is denoted as

$$adv(v_j \rightarrow v_i)_{[k]} \tag{7}$$

A *path* is a particular case of an *assembled path* in which only one element is assembled. The set of available assembled paths for a node v_i at iteration k is the set,

$$\Phi_{v_i[k]} = \{adv(v_j \rightarrow v_i)_{[k+1]}, \forall v_j \in peers(v_i)\} \tag{8}$$

The BGP-XM mapping is defined locally at node v_i as the operation of selecting the K-best paths from $\Phi_{v_i[k]} \in \mathbb{P}(\mathcal{P}(v_i, v_0))$, which can be assembled together according to the K-BESTRO algorithm (Section 3.2) and propagating the BGP-XM advertisement appropriately. The analysis does not assume that all nodes use the same K-BESTRO configuration, and the following notation is used to differentiate among K-BESTRO behaviors according to each particular configuration, so that $\mathcal{F}_{v_i}(\Phi_{v_i[k]})$ denotes the behavior at node v_i. The mapping can also be defined as the set of local operations at each vertex during the k^{th} iteration like

$$\boldsymbol{\mathcal{F}}(\boldsymbol{\Phi}_{[k]}) = \{\mathcal{F}_{v_i}(\Phi_{v_i[k]}), \quad v_i \in V\} \tag{9}$$

Before advertising them, the candidate paths are ranked. The rank value of a path is denoted as $\lambda(\theta)$ and the paths belonging to the selected set, i.e. the most preferred paths, have a ranking value of $\lambda_{max}(\Phi_{v_i[k]})$.

Given the set of advertisements in Φ_{v_i} and the ranking procedure at v_i, $\mathcal{F}_{v_i}$ (which establishes ranking values λ, and its maximum for a given set of announcements, λ_{max}), the set of most preferred paths at iteration k can be defined as,

$$\beta_{v_i[k]} = \mathcal{F}_{v_i}(\Phi_{v_i[k]}) \tag{10}$$

where,

$$\beta_{v_i[k]} = \{\theta \in \Phi_{v_i[k]} \ / \ \lambda(\theta) = \lambda_{max}(\Phi_{v_i[k]})\} \tag{11}$$

For each node v_i, its candidate set Φ_{v_i} is updated with the advertisements $adv(v_j \rightarrow v_i)_{[k+1]} \equiv \beta_{v_i,[k]}$ from the peering routers, such that the updated overall path assignment is defined like

$$\begin{aligned} \boldsymbol{\Phi}_{[k+1]} &= \{\Phi_{v_i,[k+1]}, \forall v_i \in V\} \\ &= \{adv(v_j \rightarrow v_i)_{[k+1]}, \forall v_j \in peers(v_i)\, \forall v_i, v_j \in V\} \\ &= \{\beta_{v_j,[k]}, \forall v_j \in V\} \\ &= \{\mathcal{F}_{v_j}(\Phi_{v_j[k]}), \forall v_j \in V\} \end{aligned} \tag{12}$$

and according to Eq.9, we get to the iterative equation

$$\boldsymbol{\Phi}_{[k+1]} = \boldsymbol{\mathcal{F}}(\boldsymbol{\Phi}_{[k]}) \tag{13}$$

As the synchronous execution of the mapping goes on, the paths in $\Phi_{v_i[k]}$ change dynamically until either the stable-state is reached or it continues changing infinitely. Which of these two events happen, depends on the following result.

Theorem 1. (Safety) Given a network graph $G = \langle V, E \rangle$, given the set of policies S defined by each vertex in V, a synchronous distributed Bellman-Ford mapping $\boldsymbol{\mathcal{F}}(\boldsymbol{\Theta}_{[k]})$ iterating over the path assignment $\boldsymbol{\Theta}_{[k]}$ which is initially defined as,

$$\Theta_{v_i[0]} = \begin{cases} \{v_0\}, & i = 0 \\ \emptyset, & i \neq 0 \end{cases} \tag{14}$$

If every policy relation over S is *anti-reflexive* then the mapping is able to grow the path assignment at each iteration until the fixed-point of the following equation is reached at some iteration m,

$$\boldsymbol{\Theta}_{[m]} = \boldsymbol{\mathcal{F}}(\boldsymbol{\Theta}_{[m]}) \tag{15}$$

Thus, it can be stated that in absence of reflexive policies the protocol is able to converge synchronously.

Proof: (See Appendix A for the complete proof) If at certain iteration m the algorithm is not able to progress, i.e. increase an stabilized assignment, and it has not reached a fixed point, then a reflexive relation amongst the propagated paths can be found as a consequence of the applied policies.

□

Since the execution of our protocol is not free from communication delays, besides Theorem 1, we need to guarantee the convergence under asynchronous execution of BGP-XM.

Theorem 2. (Asynchronous Convergence) Given a network graph $G = \langle V, E \rangle$ with n nodes, the set of policies S defined by each node and a distributed BGP-XM mapping $\mathcal{F}(\mathbf{\Theta}_{[k]})$ iterating over the path assignment $\mathbf{\Theta}_{[k]}$, if every policy relation over S is *anti-reflexive* then the mapping is able to asynchronously converge to a multipath assignment of paths over G.

Proof. (See Appendix B for the complete proof) Proving asynchronous convergence is equivalent to proof that any change in the network creates a sequence of sets with the *feasible* path assignments, which is decreasing in the size of the sets until only one feasible assignment is left.

□

5.3. Stable multipath policy guideline

The work of Gao and Rexford [21] studies how to define routing policies in the Internet to avoid instabilities in BGP. The resulting guidelines are based on the business relationships between ASes. Using the stability condition derived previously, one of the guidelines in [21] can be reformulated for multipath. The guideline can be proven stable if no reflexive relation can be constructed in the network.

Note that the guidelines stated by Gao and Rexford do not consider instabilities generated due to the internal distribution of routes. For example, instabilities induced by MED [25] or Route Reflector configurations [30] are not addressed. The guidelines proposed in this subsection are an equivalent for multipath to the ones presented in [21].

We do not claim that the resulting routing policy defined by this guideline is the only policy ASes can follow to achieve stability. However, this guideline covers the most common business relationships found on the Internet. The following two assumptions must hold:

Assumption 1. An AS advertises paths coming from its providers only to its customers. Paths coming from peers only to its customers and finally, paths coming from its customers to other customers, peers and providers.

Assumption 2. A customer AS cannot be an indirect provider of one of its direct providers.

Now, we present the guideline. Basically the proofs fail to construct reflexive relations, and so stability is assured, when the policies are defined according to it.

Guideline. If every AS assigns a higher local preference to paths received from its customers than to paths received from its peers, and they assign higher preference to paths coming from its peers than to paths coming from its providers, then BGP-XM is able to converge.

In addition, convergence is possible regardless of the size and characteristics of the paths in the multipath set.

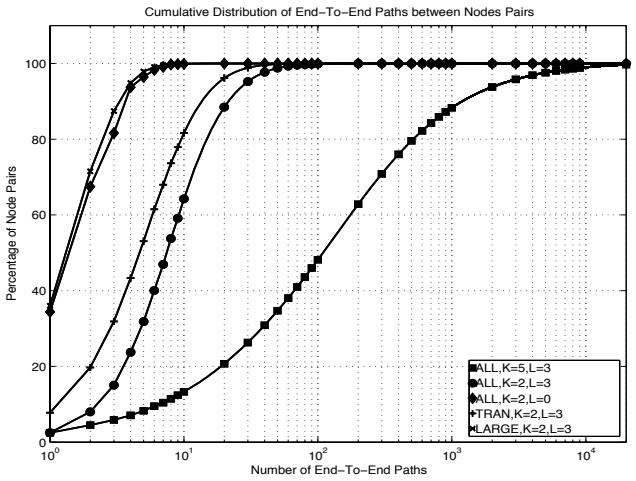


Figure 11: Cumulative End-to-End Diversity per Pair of Nodes

Proof. (See Appendix C for complete proof) The proof fails to construct a reflexive relation amogst propagated paths when the policies follow the guideline.

□

6. Performance evaluation

This section studies the performance of BGP-XM in a large-scale network. The experiments are carried out at the AS-level scope, i.e. each AS is represented as a single router. The Internet-like topology is extracted from the CAIDA Internet AS-Level Topology measurement project [31], with the AS business relationships dataset taken from [32]. The topology has a total number of 36127 ASes, divided in 84.38% of stub ASes, 0.12% of transit ASes with large connectivity, and 15.5% of transit ASes with lower connectivity. The results presented were obtained using a modified version of the C-BGP simulator [33] with support for BGP-XM[2] [34]. Throughout this section we refer to the BGP-XM configuration parameters ULMP and K-Best as the variables L and K respectively. Experiments in which BGP-XM is deployed in the whole Internet AS-topology are labelled as ALL, the label TRAN refers to experiments in which only the transit ASes deploy BGP-XM (i.e. no stubs), whereas LARGE implies that only transit ASes with large connectivity are using BGP-XM.

Path Diversity. We start the characterization of the protocol studying the path diversity disclosed by BGP-XM. The analysis presented in this section focuses on the path diversity that can be disclosed by using paths through several next ASes. Path diversity is evaluated as the amount of end-to-end paths for each pair of source and destination ASes.

The results presented here are an upper bound of the diversity that could be obtained in a real network, since the possible effects of the internal topology of the ASes and BGP configurations, further than the LOCAL_PREF, is not quantified. Furthermore, in this analysis we do not

[2]Available On-Line at `http://www.it.uc3m.es/lpgonzal/mcbgp.html`

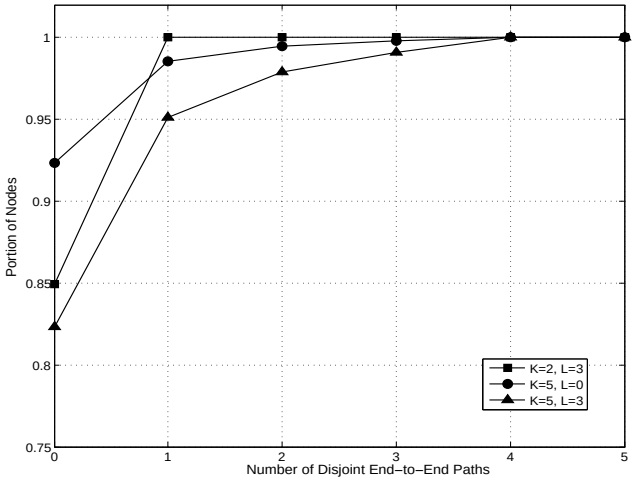


Figure 12: Cumulative Disjoint End-to-End Paths

consider the path diversity obtained from parallel interconnections between neighboring ASes. Accurate data about the redundant interconnections between two neighboring AS is hard to obtain [5, 35]. For that reason, it is not possible to include them in the analysis and in our model we consider that there is at most one link between two ASes. We have justified in previous section that BGP-XM can exploit those links in the same way as other BGP Multipath solutions [17].

Results are displayed in Fig.11, where the CDF of the end-to-end paths measured for each pair is shown. These results show that if all the ASes in the network use a multipath policy defined with $K = 5, L = 3$, a large path diversity is disclosed. For the rest of the section, we define an alternative path as a path with at least one AS number not present in the AS_PATH of the BGP best path. Out of the evaluated pairs, roughly 95% of the measured pairs have at least one alternative path. In 40% of the pairs this amount rises to 100 or more alternative paths and between $1,000$ and $10,000$ above the $90-$ *percentile* of the pairs.

The K parameter establishes the trade-off between the disclosed path diversity and the routing table growth. That effect is analyzed later on, but in Fig.11 it can be seen that it has a severe impact on the disclosed path diversity. For instance, keeping $L = 3$ and reducing K from $K = 5$ to $K = 2$, the pairs above the $90-$ *percentile*, i.e. pairs with the highest amount of paths, shifts from a diversity of $1,000 - 10,000$ to $20 - 50$ alternative paths. The case of *equal-AS_PATH-length* multipath, i.e. $L = 0$ shows even lower diversity, which points out the possible benefits of relaxing the constrains of BGP over the AS path length. Moreover, this two result, i.e. using multiple next ASes and different AS_PATH lengths, shows that the currently deployed multipath solutions by manufacturers (e.g. Cisco Multipath BGP option [17]) underuse the existing diversity.

Path Disjointness. In addition to perform a quantitative analysis of the existing path diversity, we analyze also some qualitative aspects. Many authors point out that multipath routing in general should provide better network performance. The best case for bottleneck avoidance and reliability is when the paths are disjoint [6]. We have measured the number of paths between each AS and the destination AS, which are totally disjoint (i.e. node and link disjoint) to the BGP best path.

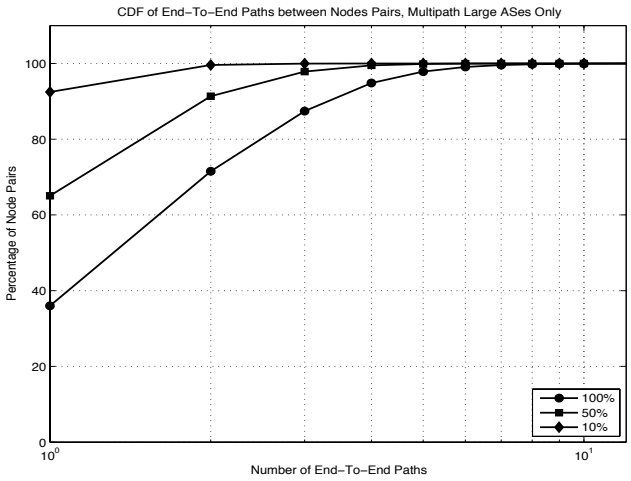


Figure 13: CDF, Impact in the Diversity of Incremental Deployment in Largely Connected ASes

Fig.12 contains the CDF of the disjoint paths for the cases where $K = 2, L = 3$ and where K and L are varied. The amount of disjointness is closely related to the path diversity disclosed, so for $K = 5, L = 3$, more pairs of nodes get more disjoint paths. The amount of AS pairs with no disjoint alternative increases for more constrained values of K and L, specially when paths are limited to have the same AS_PATH length. In any case, the number of pairs without a disjoint alternative is very large (above 80%).

We assume that this is due to the small number of nodes that form the core of the Internet. Furthermore, most end-to-end paths pass through the core of these highly connected ISPs, so the probability of coming across one of them in several paths is very high. In turn, the latter should mean that, since most paths cross the core of the network and pass through the same ASes, enabling a rich multipath solution in the downstream part of the paths, i.e. between the core and the destination AS, should have a high impact in the path diversity and therefore, core ASes (which we also refer as Tier-1) are the ideal candidates to embrace a multipath BGP solution as ours. This is analyzed in the following paragraph.

Incremental Deployment. One of the strengths of BGP-XM compared to other novel multipath inter-domain proposals is that it features incremental deployment. Once a transit AS deploys BGP-XM, its customers automatically benefit from additional end-to-end paths. Moreover, this beneficial effect is additive.

Fig. 13 shows the impact of deploying BGP-XM in large connected ASes, such as Tier-1s. Provided that we are assuming valley-free topologies [21], Tier-1s are the ideal candidate set of ASes to study the additive effect of larger BGP-XM deployments. In valley-free topologies, a Tier-1 either receives paths from one of its direct customers or through another Tier-1. Even though the number of largely connected ASes is very low, deploying BGP-XM in half of them causes that more than 30% percent of node pairs acquire at least one alternative path, as depicted in Fig. 13. Besides, deploying BGP-XM in all large Tier-1s allow the use of multiple paths in more than 60% of the node pairs and 3 or more paths above the $70 - percentile$.

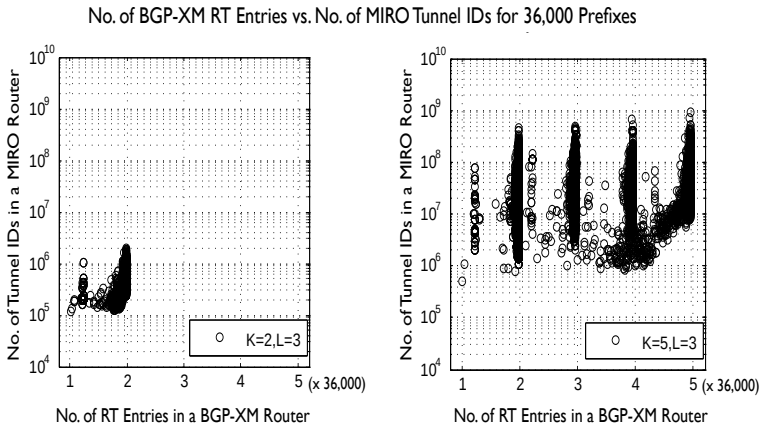


Figure 14: Scalability of Multipath Routing with BGP-XM vs. MIRO. Each marker represents the amount of routing information stored in a border router of each AS to reach the 36,000 simulated prefixes. The horizontal axis represents the number BGP-XM routing table entries (normalized to 36,000) versus the vertical axis, which represents the number of MIRO tunnel IDs to be stored so as to exploit the same path diversity as BGP-XM.

Scalability of Routing Information. We have shown the benefits of adopting BGP-XM as inter-domain multipath solution. Nevertheless, all those benefits may not justify an overhead in the routing information stored in the router, especially in the data-forwarding entries in the FIB, where the memory is probably the most expensive component in the router. Obviously, the comparison with unipath BGP is unfair, but we can compare two multipath solutions in terms of overhead. MIRO relies on a tunnel negotiation protocol to achieve larger path diversity. This implies that for each path a tunnel ID must be kept in the router to forward the packets in the tunnel. On the other hand, for a given prefix, BGP-XM just requires one entry per neighboring AS, regardless the neighboring AS has one or several paths available. The hop-by-hop class of routing of BGP-XM makes that the sequence of ASes followed after the next AS are totally transparent to the local AS.

If we display the number of tunnels to be established by a solution like MIRO versus the number of entries that BGP-XM requires in the routing table to disclose the same path diversity along the Internet, we can compare the overhead ratio between these two approaches. Fig.14 shows a scatter plot in which each datum represents an AS (with one router per AS). For each AS, we show in the horizontal axis the number of entries in the complete routing table (including the 36,000 destinations) versus the corresponding number of end-to-end tunnels that a router using MIRO should negotiate [13] to achieve the same degree of path diversity (vertical axis). Notice, that the number of entries in the routing table for BGP-XM is normalized to the number of destination. With this normalization we aimed at providing also comparison with the size of a conventional BGP router. So to speak, numbers along the horizontal axis shows how many entries a BGP-XM routing table has for each entry in a conventional BGP routing table.

Results show that BGP-XM is much more scalable than MIRO. In the worse case for BGP-XM and the best for MIRO ($K = 5, L = 3$), BGP-XM requires 5 times the routing table of unipath BGP (about 180,000 entries). While this may seem a large increment in the size of routing tables, for the same diversity MIRO needs to negotiate and store about 10^6 tunnel IDs, which seems impractical and it rises an important scalability issue.

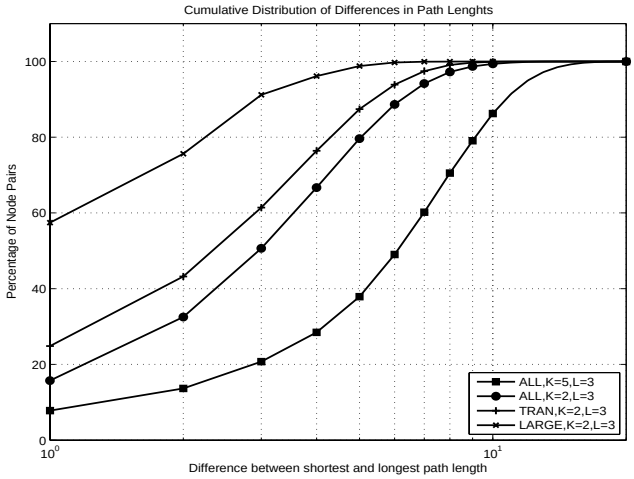


Figure 15: Cumulative End-to-End AS Path Length Difference

Impact on the AS_PATH Length and Path Stretch. Finally, we analyze the impact of using longer paths. We should expect a foregone result as a consequence of the *price of diversity* [36] (i.e. more path diversity implies longer path length). Fig. 15 shows the effect of BGP-XM in the quality of the paths. In this case the results focus on the potential increment in the AS path length due to the additional disclosed paths. Those results show the CDF of the measured difference between the shortest and the longest AS path for each pair of ASes. The impact of different values for $L = 0$ is not shown, as the effect of the L parameter in the end-to-end path length is obvious. Regarding the parameter K, the results show that it also has an impact in the AS path length. Larger path diversity also implies larger degradation of the AS path length. For instance, if $K = 5, L = 3$, 60% of the measured paths have a stretch (difference between the shortest and the longest path) that is higher than 5 ASes.

7. Related work

Several multipath inter-domain protocols have been proposed in the last years. Invariably, they encounter the problem of advertising multiple paths between domains that use BGP. In this section we collect and enumerate the different proposals according to the type of mechanism used to overcome the advertisement limitation introduced by BGP.

The novel Internet multipath architectures presented in [16, 15, 37] propose a loose version of source routing to influence the path followed by a packet. The effectiveness of those architectures increases if inter-domain paths can be also influenced. However, the inter-domain extension of those mechanisms [15] suggests that only one of the multiple available paths should be advertised among domains to indicate that the destination is reachable through that AS. Since loops may appear in this case, additional mechanisms to detect them must be used. Those proposals can benefit from BGP-XM and exploit the available path diversity without hiding relevant AS path information.

Another alternative is to propagate additional paths between ASes using a parallel mechanism in addition to the conventional BGP routing advertisement. MIRO [13] allows AS border routers

in different domains to negotiate alternative paths. The paths can be used without creating loops by means of inter-domain tunnels in the data plane. Other approaches [14] define a new BGP capability that is only understood by multipath routers. Nevertheless, the deployment of these solutions is conditioned to the coordination of at least two neighboring ASes. A single AS cannot deploy them individually, which is the main shortcoming of these approaches. In addition, the propagation of individual paths is likely to create scalability issues. As for IBGP [38], trade-offs between the scalability and the performance of those solutions exist. The results presented in section 6 show that BGP-XM features interesting scalability properties, which are suitable for large-scale deployments.

Multipath BGP extensions such as [17, 39] are the only solutions deployed in practice. Cisco routers [17] may select multiple paths with the same AS_PATH, or with the same AS_PATH length and being received from the same neighboring AS. This second case is not compatible with Cisco's so-called eiBGP feature, so it may not be available in some configurations. Even in the less restrictive case, routes with different AS_PATH may be selected only if the different routers of the neighboring AS selected their paths according to the 'prefer shortest distance to NEXT_HOP' or 'prefer EBGP over IBGP', since the rest of the rules result in a single route being selected for the whole AS. Similar requirements are stated for Juniper routers [39], allowing in this case the selection of routes from different external confederation peers. Therefore, it is not surprising to discover that multipath involving paths across different ASes is not common, as experimentally shown by the combination of AS-level information with paris-traceroute sampling [35]. Provided that, each pair of ASes negotiate the necessary agreements so as to perform coordinated inter-domain traffic engineering, as suggested in [7, 8], the effort required for an AS to balance the traffic between two or more neighboring ASes does not increase in complexity.

According to [5, 13], a large portion of the existing path diversity is dismissed by the constrains of BGP Multipath and more relaxation of the path selection rules should be beneficial. The work presented in [40] is similar to our approach in the sense that it relies on a *path aggregation* technique to advertise the multipath set to legacy routers. Thanks to the path aggregation, multipath routers can use paths with different AS_PATH attributes without creating AS-level loops, although in this solution they are still constrained to have the same length. Results in section 6 show that *unequal* AS path length multipath (i.e. ULMP) exploits even more path diversity than equal AS path length. Furthermore, the solution presented in [40] leaves many open issues, since it does not define a the path selection process, the treatment of MED attributes or the stability of the protocol. All those issues have been addressed in this paper for BGP-XM.

The idea of using the AS_PATH attribute to include information about the multiple paths selected by a router in order to perform loop detection has been presented by the authors in [41] and [42]. Our current paper present a complete definition of the mechanism aimed to ease its deployment in the current Internet, along with a stability and performance analysis.

Schapira, Zhu and Rexford [43] propose removing the AS_PATH attribute from the route selection process and aggregating the information of multiple routes into the AS_PATH by using the AS_SET attribute just to allow loop detection. Compared to this proposal, BGP-XM provides better co-existence with BGP, since it supports current traffic engineering techniques based on AS_PATH prepending and the deployment guidelines stated assure stability for an arbitrary combination of ASes deploying BGP or BGP-XM.

Balon and Leduc [44] go a step further by proposing the aggregation of multiple routes with different AS_PATH advertising just one of the selected routes (so that it does not include all the ASes that could be traversed, as BGP-XM does). In this case, information used to detect forwarding loops is lost, so they propose a guideline, inspired in the work of Gao and Rexford

[21], which results in protection against loops. The most relevant difference with BGP-XM is that the selection of multiple routes with different AS_PATH length is not allowed in [44], which highly restricts path diversity, as shown by our simulations. Aggregation of routes with different MED is not allowed either. In addition, our work is completed with a stability framework, which allows extending the analysis to other configuration guidelines.

8. Conclusions

In this paper we have addressed the design of an inter-domain multipath routing framework compatible with unipath BGP operation, which is able to disclose high path diversity. The requirement of being backwards compatible with BGP has resulted in the use of the same route update message format as BGP, the adherence to the next-hop routing multipath model (as opposed to a source-routing multipath model), and an effort to preserve the complex BGP semantics resulting from the route processing, filtering and selection processes. The task of accomplishing both goals, disclosing high path diversity and being backwards compatible with BGP, has proven to be arduous. BGP was designed to select a single route according to a set of complex sequential criteria, and the selection of multiple routes which comply with the business models defined for traffic exchange, without breaking usual traffic engineering techniques, is not an easy task. Complexity is exacerbated by the need to pack the information regarding to multiple paths in a single update. We have stated that routes with different AS_PATH values can be selected without violating usual BGP functionality, and the same occurs with routes with different ORIGIN. We have analysed the restrictions to aggregation imposed by different MED values, for aggregating EBGP and IBGP routes, and for aggregating routes with different internal costs to the NEXT_HOP.

The BGP-XM framework, the general path assembling algorithm which allows the selection of routes with different AS_PATH, and the K-BESTRO multipath selection process which relax the usual BGP selection rules, represent a complete and flexible approach to improve previous BGP multipath solutions.

Regarding to stability, we have proven that the policy relations must be *anti-reflexive* to assure convergence. This result has been used to generate configuration guidelines, derived from those proposed in [21], which are assumed to be widely used. The result stating that anti-reflexive policy relations assure asynchronous convergence can be used to derive guidelines or mechanisms under which BGP-XM is stable against oscillations induced by MED or route reflectors.

Finally, we have shown by means of simulations that BGP-XM can disclose high path diversity. We have analyzed the effect of changing the maximum number of routes selected and changing the maximum AS_PATH distance variation among the selected routes. Simulations show that deploying BGP-XM in a very small number of ASes, provided that they are largely connected, enables multipath for a large number of ASes.

As future work, we consider the study on the impact that BGP-XM may have on the BGP churn. Another area of interest lies on the analysis of the integration of BGP-XM and the Route Reflector technology, used to improve the performance of the BGP route exchange inside an AS. In addition, effort should be devoted to the design of outbound traffic engineering techniques which could exploit multiple egress points for transit ASes [45].

9. Acknowledgements

The authors Francisco Valera, Marcelo Bagnulo and Jose M. Camacho are partially funded by the European Commission by means of the Trilogy project (ICT-216372) within the 7th

Framework Programme and the Telefónica-UC3M chair in Future Internet for the Productivity. The work of Alberto García-Martínez has been partially supported by the eeCONTET project (TEC2011-29688-C02-02) granted by the Spanish Science and Innovation Ministry. Acknowledgements to Lisardo Prieto, from University Carlos III de Madrid, for his contribution to the modification of the C-BGP simulator to support BGP-XM.

Appendix A. Proof of Theorem 1

Before presenting the proof of the synchronous convergence condition for BGP-XM, we have to define the concepts of *feasible* and *stabilized* set of paths. The concept of *feasible* multipath assignment is rather intuitive if we define the set of all possible multipath assignments as the following Cartesian product (which includes partial assignments),

$$\chi = \prod_{v_i \in V} \mathbb{P}(\mathcal{P}(v_i, v_0) \cup \emptyset) \tag{A.1}$$

Therefore a multipath assignment $\mathbf{\Phi} = (\Phi_{v_0}, \Phi_{v_1}, \ldots, \Phi_{v_n}) \in \chi$ provides to each vertex v_i a set of paths Φ_{v_i} to reach the origin. In addition, for each vertex in V, we define the following

Definition 1. Given two vertex $v_i, v_j \in V$ such that $(v_i\ v_j) \in E$, the assignment Φ_{v_i} is said to be *consistent* with Φ_{v_j} if $\forall \rho \in \Phi_{v_i}$ of the form $\rho = (v_i\ v_j)\theta$, it holds that $\theta \in \Phi_{v_j}$ and $v_i \notin \theta$ (to ensure loop-freeness).

It seems clear from the definition of χ that not all the components of $\mathbf{\Phi} \in \chi$ are necessarily consistent with each other. Since our protocol handles only local information, the paths it is able to construct must be consistent for all the vertex in the path. Hence, the definition of a *feasible* multipath assignment for our protocol can be expressed as

Definition 2. A multipath assignment $\mathbf{\Phi} \in \chi$ is said to be *feasible* if $\forall v_i, v_j \in V$ and $(v_i\ v_j) \in E$ then Φ_{v_i} is consistent with Φ_{v_j}.

Before defining the concept of *stabilized* multipath assignment, the following relations between multipath feasible assignments must be defined,

Definition 3. Let $\mathbf{\Phi}, \mathbf{\Phi}' \in \chi$ be two feasible partial multipath assignments. Then, $\mathbf{\Phi}'$ contains $\mathbf{\Phi}$, i.e. $\mathbf{\Phi} \subseteq \mathbf{\Phi}'$, if $\Phi_{v_i} \subseteq \Phi'_{v_i} \quad \forall v_i \in V$ and $\mathbf{\Phi} \subsetneq \mathbf{\Phi}'$, if $\mathbf{\Phi} \subseteq \mathbf{\Phi}'$ and $\Phi_{v_i} \subsetneq \Phi'_{v_i}$ for some v_i.

Definition 4. Given a partial feasible multipath assignment $\mathbf{\Phi}$, the set $\mathbf{\Psi}(\mathbf{\Phi})$ defined as,

$$\mathbf{\Psi}(\mathbf{\Phi}) = \{\mathbf{\Phi}' \in \chi \ / \ \mathbf{\Phi} \subseteq \mathbf{\Phi}'\} \tag{A.2}$$

is the set of feasible assignments which contain the path assignment $\mathbf{\Phi}$.

Definition 5. An assignment $\mathbf{\Theta}_{[k]}$ is said to be *stabilized* if for all the sets of feasible sets containing $\mathbf{\Theta}_{[k]}$, i.e. $\forall \mathbf{\Phi} \in \mathbf{\Psi}(\mathbf{\Theta}_{[k]})$, it holds that

$$\mathbf{\Phi} \supseteq \mathbf{\Theta}_{[k]} \quad \text{implies} \quad \mathcal{F}(\mathbf{\Phi}) \supseteq \mathbf{\Theta}_{[k]} \tag{A.3}$$

The latter means that for any feasible assignment $\mathbf{\Phi}$ containing $\mathbf{\Theta}_{[k]}$, an iteration of BGP-XM over the assignment $\mathbf{\Phi}$ does not remove any path in $\mathbf{\Theta}_{[k]}$. Therefore, any path $\theta \in \mathbf{\Theta}_{[k]}$ is part of the fixed-point solution of Eq.6 for the function $\mathcal{F}$ and its ranking value verifies that

$$\lambda(\theta) = \lambda_{max}(\mathbf{\Psi}(\mathbf{\Theta}_{[\mathbf{m}]})) \quad \forall m \geq k \tag{A.4}$$

Definition 6. Let $C_{[k]}$ be the set of *converged* nodes at the k^{th} iteration. Any node $v_i \in C_{[k]}$ verifies that the set $\beta_{v_i[k]}$ belongs to the stabilized assignment at iteration k, what means that v_i converged at iteration k or before. Being $m \leq k$ the iteration at which v_i converged, then $\forall n > m$, $\beta_{v_i[n]} = \beta_{v_i[m]}$ and, therefore $adv(v_i \rightarrow w)_{[n]} = adv(v_i \rightarrow w)_{[m]}$.

Definition 7. Let $D_{[k]} \subseteq V - C_{[k]}$ be the set of nodes which are *direct peers* of the converged nodes, then, $\forall v \in D_{[k]}$, $\exists u \in C_{[k]}$ and $e = (v\ u) \in E$.

At this point and using the definitions stated above, we can define and establish the progress condition of BGP-XM in the following lemma,

Lemma 1. (Progress condition) Let S be the set of routing policies of nodes in G. If any relation among policies in S is anti-reflexive and the current overall stabilized assignment $\mathbf{\Theta}_{[k]}$ is not a fixed-point of the mapping, then there is an assignment $\mathbf{\Theta}_{[k+1]}$ such that,

1. $\mathbf{\Theta}_{[k+1]} \supsetneq \mathbf{\Theta}_{[k]}$
2. $\mathbf{\Theta}_{[k+1]}$ is also stabilized
3. $\forall \mathbf{\Phi} \in \mathbf{\Psi}(\mathbf{\Theta}_{[k]})$, then $\mathcal{F}(\mathbf{\Phi}) \supseteq \mathbf{\Theta}_{[k+1]}$

Proof: If $\mathbf{\Theta}_{[k]}$ is stabilized it means that $\mathcal{F}(\mathbf{\Theta}_{[k]}) \supseteq \mathbf{\Theta}_{[k]}$ and $\mathcal{F}_{v_0}(\mathbf{\Theta}_{[k]}) = \{v_0\}$, $k \geq 0$. In order to increase the multipath stabilized assignment there must be at least one node $v \in D_{[k]}$ peer of $u \in C_{[k]}$ such that,

1. By definition 6 u has a stabilized set $\Theta_{u[k]}$, then $\forall \theta \in \Theta_{u[k]}$ it holds $\theta \in \mathbf{\Theta}_{[k]}$
2. Given $\rho = (v\ u) \in E$, $\alpha = adv(u \rightarrow v)_{[k]}$, it holds that

$$\lambda(\rho\alpha) = \lambda_{max}(\mathbf{\Psi}(\Theta_{v[m]})) \quad \forall m \geq k \tag{A.5}$$

   hence $\rho\alpha \in \Theta_{v_i[k+1]}$ (i.e. $\rho\alpha$ is stabilized since no path with higher rank will replace it in later iterations).
3. At iteration k, $\rho\alpha \in \mathbf{\Theta}_{[k+1]} = \mathcal{F}(\mathbf{\Theta}_{[k]})$ and $\rho\alpha \notin \mathbf{\Theta}_{[k]}$

If such a node v exists then the proof of the lemma is completed since by construction $\mathcal{F}(\mathbf{\Theta}_{[k]}) \supsetneq \mathbf{\Theta}_{[k]}$. By definition 5 and Eq.A.5, $\rho\alpha$ will not be removed in further iterations, therefore $\mathcal{F}(\mathbf{\Theta}_{[k]})$ is stabilized.

Now we show that if that node $v \in D_{[k]}$ does not exist then the anti-reflexivity property does not hold over the policies in S. If v does not exist then no node $v_1 \in D_{[k]}$ is able to find a *direct* path Γ_1 constructed like above, $\Gamma_1 = \rho\alpha$, for any peer node $u \in C_{[k]}$ and being $\lambda(\Gamma_1) = \lambda_{max}(\mathbf{\Psi}(\Theta_{v[m]}))\quad \forall m \geq k$. Therefore, v_1 prefers more a path Δ_1 that is not through a converged peer. Using the *preference* and *composition* relations, we can express the policy relation between Γ_1 and Δ_1 like

$$\Delta_1 \overset{\lhd}{\approx} \Gamma_1 \tag{A.6}$$

Then, if Δ_1 is not one hop away to a converged vertex, Δ_1 must come from a propagated version of a direct path of some node $v_2 \in D_{[m]}$. Therefore, Δ_1 can be constructed like $\Delta_1 = \Pi_2\Gamma_2$. Path Π_2 is an arbitrary path passing through nodes in $V - C_{[m]}$ and $\Gamma_2 = \rho'\alpha'$, with $\rho' = (v_2\ u') \in E$ and $\alpha' = adv(u' \rightarrow v_2)_{[m]}$, is a direct path of v_2. In terms of policy relations the latter can be expressed as

$$\Gamma_2 \overset{\blacktriangleleft}{\approx} \Delta_1 \overset{\lhd}{\approx} \Gamma_1 \tag{A.7}$$

Using the same reasoning, v_2 is not choosing any direct path Γ_2, otherwise the path Δ_1 would become stabilized and the stabilized paths assignment would grow. Therefore the set $\beta_{v_2[m]}$ is formed by at least one path Δ_2 which is not direct and goes through a direct path announced by some node $v_3 \in D_{[m]}$. The same procedure is repeated for v_3 and we get to the relation

$$\Gamma_3 \overset{\blacktriangleleft}{\nsim} \Delta_2 \overset{\lhd}{\nsim} \Gamma_2 \overset{\blacktriangleleft}{\nsim} \Delta_1 \overset{\lhd}{\nsim} \Gamma_1 \tag{A.8}$$

The relation keeps repeating for every element in $D_{[m]}$ until it hits v_1 again, producing a circular relationship of policies that cannot be fulfilled simultaneously,

$$\Gamma_1 \overset{\blacktriangleleft}{\nsim} \Delta_n \overset{\lhd}{\nsim} \ldots \overset{\lhd}{\nsim} \Gamma_3 \overset{\blacktriangleleft}{\nsim} \Delta_2 \overset{\lhd}{\nsim} \Gamma_2 \overset{\blacktriangleleft}{\nsim} \Delta_1 \overset{\lhd}{\nsim} \Gamma_1 \tag{A.9}$$

The latter relation implies that Γ_1 is less preferred than a path which is composed by an arbitrary path and Γ_1, which is a contradiction. The latter completes the proof by showing that if the protocol gets stuck, then a reflexive relation exists among the policy relations.

□

Lemma 2. If a path $\theta = v_i v_{i-1} \ldots v_0$ does not appear infinitely often in the multipath set β_{v_i} of v_i, then there is an iteration k after which any path of the form $\rho\theta$ disappears from the network.

Proof: Given a vertex v_i, $\theta = \rho'\theta'$ with $\rho' = v_i v_{i-1} \ldots v_{j+1} v_j$ and $\theta' = v_j v_{j-1} \ldots v_1 v_0$, if $\theta = \rho'\theta'$ does not appear in $\beta_{v_i[m]} \quad \forall m \geq k$ it means that there is at least one node $v_j \quad 0 \leq j \leq i$, for which there is a path $\theta'' \in \mathcal{P}(v_j, v_0)$ such that $\lambda(\theta'') > \lambda(\theta')$, therefore θ' is not part of $adv(v_j \rightarrow v_{j+1})_{[k]}$ after iteration k. At iteration $k+1$ the nodes $w \in peers(v_j)$ cannot use the path θ' any longer. The process repeats at each iteration along the next hop in the path ρ' until $\rho'\theta'$ disappears. Thus, v_i cannot announce θ any longer and eventually $\rho\theta$ also disappears.

□

Lemma 3. The successive iterations of the BGP-XM mapping $\mathcal{F}(\mathbf{\Theta}_{[0]})$, $\mathcal{F}(\mathbf{\Theta}_{[1]})$, $\ldots$, $\mathcal{F}(\mathbf{\Theta}_{[k]})$, over the stabilized partial assignments reduce at each step the set of feasible path assignments $\mathbf{\Psi}$, i.e. $\mathbf{\Psi}(\mathbf{\Theta}_{[0]}) \supsetneq \mathbf{\Psi}(\mathbf{\Theta}_{[1]}) \supsetneq \cdots \supsetneq \mathbf{\Psi}(\mathbf{\Theta}_{[k]})$.

Proof: Since we are using a synchronous model, we can assume that changes made by the mapping at v_i are propagated to the peers of v_i in the next iteration. At iteration zero, the set of feasible paths is equal to the super-set $\mathbf{\Psi}(\mathbf{\Theta}_{[0]})$ whose elements are any feasible set $\mathbf{\Phi}$ defined by Eq. A.2. Since the mapping evolves by repeatedly applying Lemma 1, then all those paths with lower rank than stabilized paths in the current iteration are not announced anymore. Then, by Lemma 2 lower-ranked paths and those constructed upon them eventually disappear. In other words, following iterations of the mapping will not propagate them throughout the network and they are not feasible paths anymore. Those paths are removed from the set of feasible sets at that iteration, proving Lemma 3.

□

Proof of Theorem 1: Combining the three lemmas presented in this section, we can prove Theorem 1. By applying Lemma 1 at each iteration, in absence of conflicting policy relations, the mapping is always able to increase the path assignment with at least one path such that the new assignment $\mathcal{F}(\mathbf{\Theta}_{[k]}) \supsetneq \mathbf{\Theta}_{[k]}$ is also stabilized. By Lemma 3, as the mapping is consolidating stabilized paths at each vertex, the set of feasible paths $\mathbf{\Psi}$ is decreasing, until the highest ranked paths feasible at each node are announced. Hence, there is one iteration k at which the only feasible set of paths at a certain node v_i is the set $\Phi_{v_i[k]} \in \mathbb{P}(\mathcal{P}(v_i, v_0))$ formed by elements that verify the equation $\lambda(\theta) = \lambda_{max}(\mathbf{\Psi}(\Phi_{v_i[m]}))$, $\forall\theta \in \Phi_{v_i[k]}$ and $\forall m \geq k$. Since the mapping does not

remove paths from a stabilized assignment and it cannot find higher ranked paths at any node, the next iteration $k+1$ will have as outcome the same path assignment. Therefore, we can say that the fixed-point has been hit at iteration k.

□

Appendix B. Proof of Theorem 2

According to the general results in [46], it is possible to ensure convergence for a totally asynchronous distributed fixed-point iteration if

1. The propagation of information happens *infinitely often*. In other words, it can be assumed that after a certain time $t' > t$ all the announcements $adv(v_j \rightarrow v_i)_{[t]}$ have been propagated and renewed at peer nodes.
2. *Synchronous condition*: The protocol creates at each iteration $k = 0, 1, \ldots, m$, a sequence of sets
$$X_{[0]} \supsetneq X_{[1]} \supsetneq \cdots \supsetneq X_{[n-1]} \supsetneq X_{[n]} \supsetneq \ldots \tag{B.1}$$
and it holds
$$\mathcal{F}(x) \in X_{[k+1]}, \forall x \in X_{[k]} \tag{B.2}$$
3. *Box condition*: For each iteration $k = 0, 1, \ldots, m$ and each node v_i, $i = 0, 1, \ldots, n$, there exist sets of elements $X_{v_i[k]}$ such that the set of elements $X_{[k]}$ can be expressed as the Cartesian product,
$$X_{[k]} = \prod_i X_{v_i[k]} \tag{B.3}$$
The box condition implies that different elements in $X_{v_i[k]}$ can be exchanged without affecting the final result of the iteration, i.e. the order in which the Bellman-Ford mapping allocates paths does not affect to the evolution of the mapping.

In order to prove the asynchronous convergence of BGP-XM, we need to show that the three conditions state above hold. The condition (1) is guaranteed since BGP-XM uses a reliable transport protocol to exchange the information and every node advertises its neighbors with every change in the selected multipath set. Condition (2) is guaranteed by Theorem 1. In addition, by replacing $X_{[k]}$ by $\mathbf{\Psi}(\mathbf{\Theta_{[k]}})$, the set of feasible sets containing $\mathbf{\Theta_{[k]}}$, both Eq.B.1 and B.2 can be rewritten as follows. Lemma 3 proves that the protocol creates the sequence

$$\mathbf{\Psi}(\mathbf{\Theta}_{[0]}) \supsetneq \mathbf{\Psi}(\mathbf{\Theta}_{[1]}) \supsetneq \cdots \supsetneq \mathbf{\Psi}(\mathbf{\Theta}_{[k]}) \ldots \tag{B.4}$$

which can be easily identified with the sequence in Eq.B.1. Moreover, it can be stated by definition 4,

$$\mathbf{\Theta}_{[k]} \subseteq \mathbf{\Phi}, \forall \mathbf{\Phi} \in \mathbf{\Psi}(\mathbf{\Theta}_{[k]}) \tag{B.5}$$

and by definition 5, applying an iteration of the algorithm on both sides, if $\mathbf{\Theta_{[k]}}$ is stabilized then

$$\mathcal{F}(\mathbf{\Theta}_{[k]}) \subseteq \mathcal{F}(\mathbf{\Phi}) \Rightarrow \mathbf{\Theta}_{[k+1]} \subseteq \mathcal{F}(\mathbf{\Phi}) \tag{B.6}$$

again, by definition 4,

$$\mathcal{F}(\mathbf{\Phi}) \in \mathbf{\Psi}(\mathbf{\Theta_{[k+1]}}), \forall \mathbf{\Phi} \in \mathbf{\Psi}(\mathbf{\Theta}_{[k]}) \tag{B.7}$$

so that Eq.B.2 can be rewritten as well identifying $X_{[k]} \equiv \mathbf{\Psi}(\mathbf{\Theta_{[k]}})$ and $\mathbf{\Phi} \equiv x$.

Finally, in order to complete the proof of Theorem 2, the box condition must be verified. The stabilized assignment can be also expressed like $\mathbf{\Theta}_{[k]} = (\Theta_{v_0[k]}, \Theta_{v_1[k]}, \ldots, \Theta_{v_n[k]})$. Lemma 1 guarantees that at least one node v_i that increases its feasible assignment, so that there is a set $\Phi'_{v_i[k]}$ such that $\Theta_{v_i[k]} \subsetneq \Phi'_{v_i[k]}$. As there can be more than one, the following super-set $\mathbf{\Psi}_{v_i[k]}$ can be defined as the set of feasible assignments that contain the stabilized assignment $\Theta_{v_i[k]} \subsetneq \Phi^{(p)}_{v_i[k]}$, i.e. $\mathbf{\Psi}_{v_i[k]} = \{\Phi^{(p)}_{v_i[k]}, \forall p\}$. Provided that all the assignments within the super-sets, $\mathbf{\Psi}_{v_i[k]}, \forall i$, are feasible, therefore the set of feasible sets containing $\mathbf{\Theta}_{[k]}$, can be rewritten as the following Cartesian product

$$\mathbf{\Psi}(\mathbf{\Theta_{[k]}}) = \mathbf{\Psi}_{v_0[k]} \times \mathbf{\Psi}_{v_1[k]} \times \cdots \times \mathbf{\Psi}_{v_n[k]} \tag{B.8}$$

which can be arranged to resemble Eq.B.3 as follows

$$\mathbf{\Psi}(\mathbf{\Theta_{[k]}}) = \prod_{0 \le i \le n} \mathbf{\Psi}_{v_i[k]} \tag{B.9}$$

and the box condition is proven. Provided that the three conditions are verified for the protocol, by the *General Asynchronous Convergence Theorem* (Proposition 2.1 in [46]) it can be stated that the protocol is able to converge asynchronously.

□

Appendix C. Proof of the Multipath Stability Guideline

The proof fails to construct the reflexive relation in Eq.3. Without loss of generality, the proof refers to the scenario in Fig.6. First we assume that Γ_1 comes from a customer of v_1, then according to Assumption 1, Δ_1 must come from another customer, otherwise it cannot have higher preference. Then Δ_1 is of the form $\Delta_1 = \Pi_2\Gamma_2$ (in Fig.6, Π_2 is just a link between v_1 and v_2 but in general it is an arbitrary path). Since v_1 is a provider of v_2, according to Assumption 1, the latter can only advertise paths from its customers to v_1 (notice that if intermediate nodes between v_1 and v_2 exist, the situation is the same). If Δ_2 has higher preference than Γ_2 and v_2 is following the Assumption 1, then it must come from another customer of v_2. The same reasoning applies for v_3. Now, at v_4, the path Δ_4 through v_1 should be preferred over the path Γ_4. This can only happen if Δ_4 comes from a customer of v_4, however if v_1 is a customer of v_4 and v_4 is in the chain of customers from v_1, it means that Assumption 2 is broken.

In the second case, we assume that Γ_1 comes from a peer of v_1. Therefore, Δ_1 must either come from a peer or a customer of v_1 (Assumption 1). In both cases, it means that Γ_2 and Δ_2 come from customers of v_2, otherwise they cannot be advertised to a peer or a provider. The chain of customers continue until v_4. A reflexive relation would be constructed if v_1 is a customer of v_4. If v_2 is a peer of v_1, then Δ_1 cannot be advertised to v_4 as it is a v_1 provider. If v_2 is a customer of v_1 we are in the previous case.

In the last case, v_1 learns Γ_1 from a provider, then v_2 can be a customer, peer or provider of v_1. If v_2 is a provider of v_1 and Γ_2 and Δ_2 are advertised to v_1 since they come from customers or peers of v_2, the chain continues like in the two previous cases. Otherwise, if Γ_2 and Δ_2 come from providers of v_2, then the chain of providers continue to v_4. If Γ_4 comes from a customer of v_4, then according to Assumption 1 Δ_4 must come from a customer, however v_1 does not advertise its provider v_4 with paths from other providers, therefore Δ_4 is not announced and the reflexive relation is broken. It is the same if Γ_4 comes from a peer. Only if Γ_4 comes from a provider, Δ_4 can be more preferred, therefore if v_1 is the provider of v_4 then Δ_4 is advertised, however in

that case v_4 becomes the indirect provider of one of its direct providers (through v_3 and v_2) and Assumption 2 is broken.

□

References

[1] P. Oppenheimer, Top-Down Network Design (2nd Edition), Cisco Press, 2004.

[2] D. Alderson, L. Li, W. Willinger, J. Doyle, Understanding internet topology: principles, models, and validation, Networking, IEEE/ACM Transactions on 13 (2005) 1205 – 1218.

[3] D. Meyer, University of oregon route views archive project, at http://archive. routeviews. org (2011).

[4] A. Dhamdhere, C. Dovrolis, The internet is flat: modeling the transition from a transit hierarchy to a peering mesh, in: CoNEXT.

[5] P. Merindol, V. V. den Schrieck, B. Donnet, O. Bonaventure, J.-J. Pansiot, Quantifying ases multiconnectivity using multicast information, in: Proc. ACM USENIX Internet Measurement Conference (IMC), pp. 370–376.

[6] A. Akella, B. M. Maggs, S. Seshan, A. Shaikh, On the performance benefits of multihoming route control, IEEE/ACM Trans. Netw. 16 (2008) 91–104.

[7] R. Mahajan, D. Wetherall, T. Anderson, Towards coordinated interdomain traffic engineering, Proc. SIGCOMM Workshop on Hot Topics in Networking (2004).

[8] S. Secci, J.-L. Rougier, A. Pattavina, F. Patrone, G. Maier, Peering equilibrium multipath routing: a game theory framework for internet peering settlements, IEEE/ACM Trans. Netw. 19 (2011) 419–432.

[9] N. Feamster, J. Borkenhagen, J. Rexford, Guidelines for interdomain traffic engineering, ACM SIGCOMM Computer Communication Review 33 (2003) 19–30.

[10] C. Villamizar, R. Chandra, R. Govindan, Rfc 2439: Bgp route flap damping, Internet Engineering Task Force (1998).

[11] J. He, J. Rexford, Toward internet-wide multipath routing, Network, IEEE 22 (2008) 16–21.

[12] Y. Wang, M. Schapira, J. Rexford, Neighbor-specific BGP: more flexible routing policies while improving global stability, in: Proceedings of the eleventh international joint conference on Measurement and modeling of computer systems, ACM, pp. 217–228.

[13] W. Xu, J. Rexford, MIRO: multi-path interdomain routing, in: Proceedings of the 2006 conference on Applications, technologies, architectures, and protocols for computer communications, ACM, pp. 171–182.

[14] D. Walton, et al., Advertisement of Multiple Paths in BGP (draft-walton-bgp-add-paths-06. txt), Network Working Group Internet Draft (2008).

[15] M. Motiwala, M. Elmore, N. Feamster, S. Vempala, Path splicing, ACM SIGCOMM Computer Communication Review 38 (2008) 27–38.

[16] X. Yang, D. Wetherall, Source selectable path diversity via routing deflections, in: SIGCOMM, ACM, pp. 159–170.

[17] Cisco, Bgp best path selection algorithm, `http://www.cisco.com/en/US/tech/tk365/technologies_tech_note09186a0080094431.shtml`, 2012.

[18] P. Faratin, D. Clark, S. Bauer, W. H. Lehr, P. Gilmore, A. Berger, The growing complexity of internet interconnection, Communications & Strategies 1 (2008) 51–72.

[19] M. Caesar, J. Rexford, BGP routing policies in ISP networks, Network, IEEE 19 (2005) 5–11.

[20] T. Griffin, F. Shepherd, G. Wilfong, The stable paths problem and interdomain routing, IEEE/ACM Transactions on Networking (TON) 10 (2002) 232–243.

[21] L. Gao, J. Rexford, Stable Internet routing without global coordination, IEEE/ACM Transactions on Networking (TON) 9 (2001) 681–692.

[22] C. Chau, Policy-based routing with non-strict preferences, in: Proceedings of the 2006 conference on Applications, technologies, architectures, and protocols for computer communications, ACM, pp. 387–398.

[23] Y. Rekhter, T. Li, S. Hares, RFC 4271: a Border Gateway Protocol 4 (BGP-4), Internet Engineering Task Force (2006).

[24] K. Foster, Application of bgp communities, Cisco Internet Protocol Journal (IPJ) 6 (2003) 2–9.

[25] T. Griffin, G. T. Wilfong, Analysis of the med oscillation problem in bgp, in: Proceedings of the 10th IEEE International Conference on Network Protocols, ICNP '02, IEEE Computer Society, Washington, DC, USA, 2002, pp. 90–99.

[26] J. Rexford, Handbook of optimization in telecommunications, chap, Route optimization in IP networks (2006).

[27] B. Donnet, O. Bonaventure, On bgp communities, SIGCOMM Comput. Commun. Rev. 38 (2008) 55–59.

[28] Y. Wang, I. Avramopoulos, J. Rexford, Design for configurability: rethinking interdomain routing policies from the ground up, Selected Areas in Communications, IEEE Journal on 27 (2009) 336–348.

[29] T. McGregor, S. Alcock, D. Karrenberg, The RIPE NCC internet measurement data repository, in: Passive and Active Measurement, Springer, pp. 111–120.

[30] A. Basu, C.-H. L. Ong, A. Rasala, F. B. Shepherd, G. Wilfong, Route oscillations in i-bgp with route reflection, SIGCOMM '02, ACM, New York, NY, USA, 2002, pp. 235–247.
[31] IPv4 Routed /24 Topology Dataset, Technical Report, CAIDA, 2011.
[32] X. Dimitropoulos, D. Krioukov, G. Riley, K. Claffy, Revealing the autonomous system taxonomy: The machine learning approach, in: Passive and Active Measurement (PAM) Workshop.
[33] B. Quoitin, S. Uhlig, Modeling the routing of an autonomous system with c-bgp, Network, IEEE 19 (2005) 12–19.
[34] J. M. Camacho, L. Prieto, F. Valera, Evaluation framework for adaptive multi-path inter-domain routing protocols, International Journal of Adaptive, Resilient and Autonomic Systems (IJARAS) 2 (2011) 24–44.
[35] E. Elena, J.-L. Rougier, S. Secci, Characterisation of as-level path deviations and multipath in internet routing, in: NGI, pp. 1–7.
[36] M. Pióro, D. Medhi, S. O. service), Routing, Flow, and Capacity Design in Communication and Computer Networks, Elsevier/Morgan Kaufmann, 2004.
[37] H. Kaur, S. Kalyanaraman, A. Weiss, S. Kanwar, A. Gandhi, Bananas: An evolutionary framework for explicit and multipath routing in the internet, ACM SIGCOMM Computer Communication Review 33 (2003) 277–288.
[38] V. Van den Schrieck, P. Francois, O. Bonaventure, BGP add-paths: the scaling/performance tradeoffs, Selected Areas in Communications, IEEE Journal on 28 (2010) 1299–1307.
[39] Juniper, Configure bgp to select multiple bgp paths, `http://www.juniper.net/techpubs/software/junos/junos53/swconfig53-ipv6/html/ipv6-bgp-config29.html`, 2012.
[40] J. Halpern, M. Bhatia, P. Jakma, Advertising equal cost multi-path routes in bgp, Draft-bhatia-ecmp-routes-in-bgp-02. txt (2006).
[41] T. M. Philip Eardley, D7 overall architecture including design principles, in: Deliverable, Trilogy Project.
[42] B. Ramamurthy, G. Rouskas, K. Sivalingam(eds.), Next-Generation Internet Architectures and Protocols, Cambridge University Press, 2011.
[43] M. Schapira, Y. Zhu, J. Rexford, Putting bgp on the right path: a case for next-hop routing, in: Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks, Hotnets-IX, ACM, New York, NY, USA, 2010.
[44] S. Balon, G. Leduc, Can forwarding loops appear when activating ibgp multipath load sharing?, Sustainable Internet (2007) 213–225.
[45] T. Bressoud, R. Rastogi, M. Smith, Optimal configuration for bgp route selection, in: INFOCOM 2003, volume 2, IEEE, pp. 916–926.
[46] D. Bertsekas, J. Tsitsiklis, Parallel and distributed computation, Old Tappan, NJ (USA); Prentice Hall Inc., 1989.