Loop-freeness in multipath BGP through propagating the longest path

Master's thesis

Iljitsch van Beijnum

UC3M
Departamento de
Ingeniería Telemática

Abstract—The concurrent use of multiple paths through a communications network has the potential to provide many benefits, including better utilization of the network and increased robustness. A key part of a multipath network architecture is the ability for routing protocols to install multiple routes over multiple paths in the routing table. In this paper we propose changes to local BGP processing that allow a BGP router to use multiple paths concurrently without compromising loop-freeness.

Index Terms—Routing, BGP, multipath, loop-freeness

I. INTRODUCTION

USING multiple paths concurrently to send packets to a single destination has a number of advantages. In a multipath-aware network, there are fewer occasions where available links remain unused because no traffic is routed over them. It also reduces the need to manually optimize traffic flow (*traffic engineering*). Additional benefits can be gained if transport protocols can be made aware of the multiple paths and direct flows or sub-flows over different paths. In that case, users gain better robustness because the reaction to failures for a subset of available paths to a destination can be handled at transport time scales, which tend to be much shorter than routing time scales, especially in the case of inter-domain routing [1]. It also allows for dynamic adjustment to congestion [1].

However, in order to make use of multiple paths, it is necessary for routing protocols to be multipath-aware. Common Internet Protocol routing protocols already support the use of multiple paths in either their design or implementation. OSPF [2] supports equal-cost multipath; EIGRP [3] is capable of utilizing multiple paths with differing costs. Although the specification does not support this, many BGP [4] implementations, such as those from Cisco and Juniper, are capable of utilizing multiple equal-cost paths concurrently. But since the BGP protocol as currently defined does not support multipath, this either leads to a risk of routing loops in autonomous systems downstream from the AS utilizing multiple paths, or the risk that loop-free paths are not considered eligible for multipath use.

We propose changes to BGP's path selection and path dissemination rules that allow for the use of a wide selection of paths concurrently without compromising loop-freeness. Because a router running BGP receives multiple paths to the same destination from different neighboring routers, it can select, based on policy criteria, a subset of the received paths for concurrent use. We then disseminate the path with longest AS_PATH length to downstream ASes. Although disseminating a path that has a larger number of ASes in its AS_PATH seems counterintuitive, it has the property that it allows the router to use *all* paths with a smaller or equal AS_PATH length without risking loops.

However, this change has the implication that there is no longer a one-to-one relationship between the path(s) that packets follow through the network and the path that is advertised in BGP. The resulting obfuscation of the network's topology as seen by observers at the edge of the network can either be considered harmful, for those who want to study networks or apply policy based on the presence of certain intermediate domains, or useful, for those intent on hiding the inner workings of their network. We limit ourselves to the situation where an individual BGP router locally balances traffic over multiple paths, without changing BGP semantics. This means that the changes can be incrementally deployed on individual routers which then gain multipath benefits without requiring changes in either upstream or downstream BGP routers.

We will first provide an overview of the relevant BGP path selection rules, then outline the modifications to BGP, after that prove loop-freeness, address convergence and finally briefly evaluate the result of these changes.

II. THE BORDER GATEWAY PROTOCOL

For the past 15 years, BGP-4 has been the inter-domain routing protocol used for the internet [17]. BGP is an exterior gateway protocol (EGP) that runs between routing domains, unlike interior gateway protocols (IGPs) such as OSPF, RIP, EIGRP and IS-IS that are designed to be used within a routing domain under a single administrative control. These

routing domains are called autonomous systems in BGP and are distinguished by an AS number. However, customers that do not run BGP themselves, which they do not need to do if they use a single internet service provider to connect to the internet, are part of their service provider's AS, regardless of whether the service provider administers any routers in the customer's network.

BGP is classified as a *path vector* routing protocol, closely related to distance vector protocols, which include RIP and EIGRP. The addition of a path in BGP allows for better detection of routing loops: whenever a router sees the network's own autonomous system (AS) number in a path, the router assumes the path is looping, and rejects it. So unlike distance vector protocols, BGP does not depend on a distance metric to detect loops.

Additionally, unlike interior routing protocols such as RIP, EIGRP and OSPF, BGP allows network operators to express policy. This is essential for an inter-domain routing protocol: the shortest path between two ISPs may be through a mutual customer. However, using a path through a customer like this violates the business relationship between the customer and its ISPs, so the inter-domain routing protocol must be configured to ignore this path. BGP allows for these policies and more complex ones that prefer cheap paths and fall back to more expensive ones when cheaper paths become unreachable.

Conceptually, the BGP protocol selects a best path by computing a degree of preference for all valid paths to a given destination received from BGP speakers in neighboring routing domains, and then selecting the path with the highest degree of preference (expressed as the LOCAL_PREF attribute). The BGP specification does not mandate a function or algorithm for computing the degree of preference; in practice, the preference is derived from administratively configured policy rules: the main way to configure policies is to create filters that selectively set the LOCAL_PREF attribute or another BGP path attribute to a higher or lower value for paths that match certain criteria. During BGP path selection process, the path or paths with the highest LOCAL_PREF are selected by BGP to be put in the routing table and be used to forward packets.

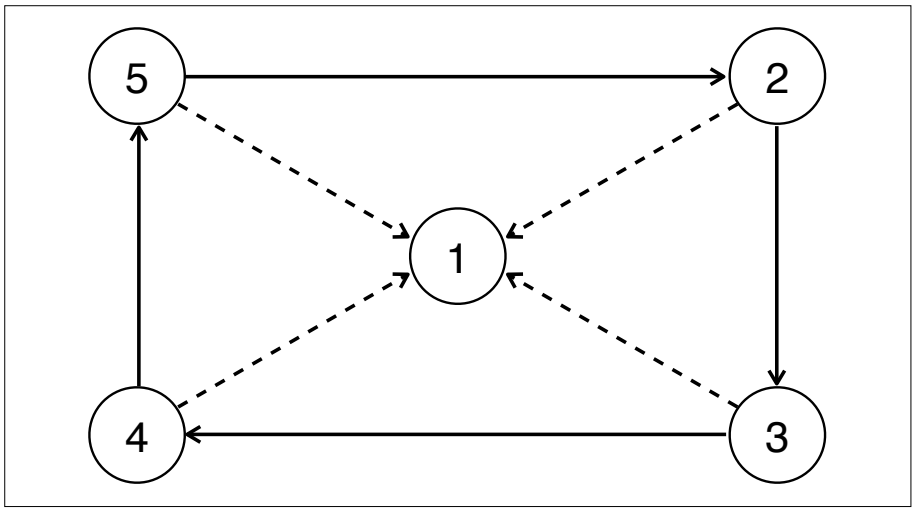

Fig. 1. Conflicting BGP policies.

An unfortunate side effect of BGP's policy support is that it allows network operators to configure conflicting policies. For instance, consider figure 1. ASes 2, 3, 4 and 5 all prefer to reach AS 1 through a longer path (as indicated by the solid arrow) rather than the direct path (dotted arrow). It's not possible to accommodate the policies of each AS. The state that BGP converges to depends on the order in which the connections become available. So

even though BGP stops the convergence process at that point, it never reaches a *stable* converged state. There are even some combinations of topology and policies which prevent BGP from converging altogether.

BGP has two modes of operation: external (eBGP) and internal (iBGP). Sessions towards BGP neighbors in autonomous systems other than the local AS are eBGP sessions; sessions towards BGP neighbors in the same autonomous system are iBGP sessions. The rules for eBGP and iBGP differ slightly. For instance, in eBGP the NEXT_HOP attribute is normally updated, but in iBGP the NEXT_HOP attribute is communicated as-is. Also, all BGP routers in an AS must maintain iBGP sessions with each other in a full mesh so paths are always propagated over iBGP directly from the router that learns them over eBGP towards all other routers in the AS. This is necessary to avoid loops because the AS_PATH attribute cannot prevent iBGP loops as the AS_PATH is not updated when a path is propagated over iBGP.

III. Valley-freeness

In [18], Gao introduces the concept of *valley-freeness*. In the valley-free model, there are three types of relationships between autonomous systems: a provider-customer relationship, a peer relationship and a sibling relationship. The provider-customer relationship entails that the provider provides connectivity to the entire internet to the customer, and the customer's customers, if any. This is a common common relationship, either between commercial ISPs and their customers, or academic or governmental service providers and their users.

In a peer relationship, the two ASes exchange routing information, and therefore packets, that have one AS (or a customer of that AS) as its source and the other AS (or a customer of that AS) as its destination. This typically happens between internet service providers (ISPs) of roughly equal size without money changing hands. In a sibling relationship, each AS provides the other AS with backup connectivity. This relationship may occur if two ASes have the same owner or have a close relationship of another kind.

The type of relationship between two ASes is determined by the policy filters each AS configures: each AS either propagates all prefixes that it knows about, or only those owned by the AS itself and its direct or indirect customers. The four permutations are listed in table 1.

AS 1 disseminates	AS 2 disseminates	Relation between ASes
Own/customer prefixes	Own/customer prefixes	Peers
Own/customer prefixes	All prefixes	Customer → ISP
All prefixes	Own/customer prefixes	ISP → customer
All prefixes	All prefixes	Siblings

Table 1. Relationship types resulting from prefix dissemination policies.

Policies and the paths through the network that they allow are valley-free if a path only has a single peering connection in it, and after the peering connection, the only other connections in the path are provider-customer links in the provider-to-customer direction or sibling links. After a provider-customer link, no customer-provider links are allowed. In figure 2, the path from A to B through ISPs (providers) 2, 1 and 4 is valley-free, but the path through ISPs 2, 3 and 4 is not.

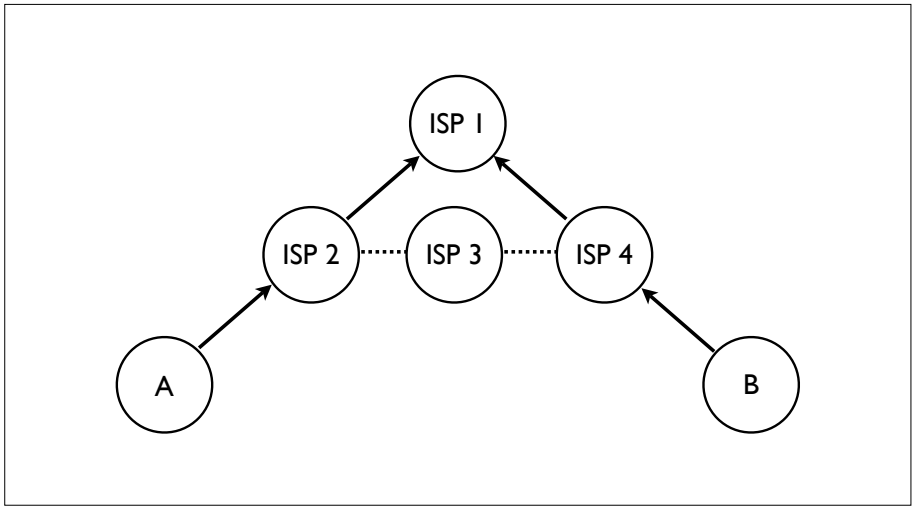

Fig. 2. Relationships between providers (ISPs) and customers for the purpose of evaluating valley-freeness.

It is widely understood that only valley-free paths can be assumed to be correct. Non-valley-free paths are possible, but they leave one or more ASes in the path without compensation for carrying traffic. For instance, in the non-valley-free path A-2-3-4-B in figure 2, ISP 3 carries traffic from A to B, but ISP 3 is not compensated for this because neither AS A nor AS B is a direct or indirect customer of ISP 3. Non-valley-free paths are regularly observed, but the majority of those is the result of configuration errors [19]. Little is known about the how and why of non-valley-free paths that are intentionally present in the internet global routing table.

IV. Gao-Rexford guidelines

Gao and Rexford [8] have shown that adopting a number of guidelines that are similar to the valley-freeness requirements and thus compatible with normal business relationships will make BGP provably converge to a stable state.

Gao and Rexford first formulate Guideline A, which applies to ASes with peer-to-peer and ISP-to-customer relationships. Guideline A requires that the LOCAL_PREF for paths learned from customers must have a higher LOCAL_PREF than those learned from peers. If each AS conforms to Guideline A, BGP will provably converge to a stable state.

They subsequently relax this requirement when Assumption P is fulfilled. This assumption groups ASes that peer together in clusters (a non-peering AS is a cluster of its own) and assumes that the provider-to-costumer relationships between clusters form a directed acyclic graph with no self cycles. In other words, when following the relationships down the provider-customer hierarchy, it's not possible to return to a place higher in the hierarchy.

With Assumption P in place, Guideline A can be relaxed as Guideline B, which mandates that paths learned from customers have an equal or higher LOCAL_PREF than paths learned from peers. Paths learned from customers are still required to have a higher LOCAL_PREF than paths learned from providers.

Guideline C further allows for backup relationships. However, all ASes must assign a fixed LOCAL_PREF, which is lower than that used for any other paths, to paths that traverse a backup link. Gao and Rexford suggest flagging such paths with a BGP community [20] so backup paths may be recognized by third party ASes (i.e., ASes that do not directly partake

in the backup relationship). However, there is no well-known community [21].

None of the guidelines disallow the non-valley-free path A-2-3-4-B in figure 2; the set of topologies conforming to the Gao-Rexford guidelines is a superset of the set of valley-free topologies.

V. Current tie-breaking rules in BGP

The BGP protocol does not accommodate using more than one path to reach a given destination, so when there is more than a single valid path with the highest degree of preference, seven tie breaking rules are applied in succession until a single path remains. These rules can be described using the notation listed in table 2.

π	the set of paths towards a destination disseminated to the local router by neighboring routers
P	the set of paths towards a destination that are under consideration for being used $P \subseteq \pi$
P^e	paths to a destination learned from neighbors in adjacent routing domains (eBGP) $P^e \subseteq P$
P^i	paths to a destination learned from neighbors in the local routing domain (iBGP) $P^i \subseteq P$
R	the set of neighboring routers
p_r	the path selected for dissemination (to router $r \in R$)
a_p	AS_PATH length for path p
s_p	the neighbouring AS from which path p was learned
l_p	LOCAL_PREF for path p
m_p	MULTI_EXIT_DISC for path p
o_p	ORIGIN for path p
h_p	NEXT_HOP for path p
b_p	the BGP identifier for r_p
r_p	address of the neighbour from which path p was learned
$d(p)$	the destination of path p
$cp(p)$	the cost to reach a destination through path p
$cp_r(p)$	the cost to reach a destination through path p that is reported to other routers
$c(x)$	the cost to reach destination x
$c_r(x)$	the cost to reach destination x that is reported to other routers
$set(p)$	TRUE if the AS_PATH of p contains an AS_SET, FALSE otherwise

Table 2. Notation.

The tie breaking rules can be expressed as follows.

a. Remove all paths that do not have the shortest AS_PATH:

$$a_p > a_q \Rightarrow \{ P \} \setminus p \qquad \forall\, p \in P, \forall\, q \in \mathrm{P}, p \neq q \quad (1)$$

b. Remove all paths that do not have the lowest ORIGIN:

$$o_p > o_q \Rightarrow \{ P \} \setminus p \qquad \forall\, p \in P, \forall\, q \in \mathrm{P}, p \neq q \quad (2)$$

c. From each subset of paths learned from the same neighboring AS, remove all paths that do not have the lowest MULT_EXIT_DISC:

$$s_p = s_q \wedge m_p > m_q \Rightarrow \{ P \} \setminus p \qquad \forall\, p \in P, \forall\, q \in P, p \neq q \quad (3)$$

d. If at least one path is learned over eBGP, remove all paths learned over iBGP:

$$P^e \neq \varnothing \Rightarrow \{ P \} \setminus P^i \quad (4)$$

e. Remove all paths that do not have the lowest interior cost towards their NEXT_HOP:

$$c(h_p) > c(h_q) \Rightarrow \{ P \} \setminus p \qquad \forall\, p \in P, \forall\, q \in \mathrm{P}, p \neq q \quad (5)$$

f. Remove all paths that are not learned from the neighbor with the lowest BGP identifier:

$$b_p > b_q \Rightarrow \{ P \} \setminus p \qquad \forall\, p \in P, \forall\, q \in \mathrm{P}, p \neq q \quad (6)$$

g. Remove all paths that are not learned from the neighbor with the lowest IP address:

$$r_p > r_q \Rightarrow \{ P \} \setminus p \qquad \forall\, p \in P, \forall\, q \in P, p \neq q \quad (7)$$

VI. Multipath Modifications to BGP

This section outlines the proposed processing rules necessary to achieve loop-free BGP multipath support. Potentially, for every possible destination, a router learns a path to that destination from each neighboring BGP router. However, BGP specifically employs the tie breaking rules to end up with a single path towards each destination when there are multiple paths with a same LOCAL_PREF value. In order to utilize multiple paths towards a destination, we follow the BGP path selection rules, in particular the rule that only paths that share the highest LOCAL_PREF are selected, up to the tie breaking rules, which we remove. However, there are additional rules as outlined below that govern which paths $p \in P$ may be used to forward packets, and more rules that determine which of those paths is disseminated to neighbors.

A. Too Many Paths

Removing the tie breaking rules has the potential to create a set of usable paths that is too large to be workable. However, many of the expected benefits can be achieved with a small number of choices [5]. Depending on hardware limitations, it may be desirable to limit the number of paths by executing the tie breaking rules until the number of paths meets a predetermined maximum α.

B. Low-quality Paths

Although the AS_PATH length is not an accurate metric of a path's quality, completely disregarding the AS_PATH length may result in selecting inferior paths, as paths with very long AS_PATHs do tend to be inferior to those with short AS_PATHs. But only accepting paths that have an equal AS_PATH length limits the number of usable paths without good reason. Also, since we need to disseminate the path with the longest AS_PATH to

downstream ASes, selecting paths with very long AS_PATHs will lead upstream ASes to prefer alternate downstream ASes, which would be detrimental for commercial network operators.

Considering this, we will use value β as the difference in AS_PATH length that is allowed between the path with the shortest AS_PATH and the path with the longest AS_PATH. For example, if β is 1 and the shortest AS_PATH among the paths in *P* is 2 ASes, then paths with an AS_PATH length of 3 will be accepted in *P* but not paths with an AS_PATH length of 4. We will evaluate different choices for β in the evaluation section.

Paths with AS_PATHs that are too long are removed from *P*:

$$a_p > a_q + \beta \Rightarrow \{ P \} \setminus p \qquad \forall\, p \in P, \forall\, q \in P, p \neq q \quad (8)$$

C. Avoiding Suppression of eBGP Paths Towards iBGP Neighbors

A central notion to BGP is that a router only disseminates the path that it uses itself for forwarding packets. With iBGP, there is the additional limitation that a router may only disseminate a path that is learned over eBGP or generated locally. Disseminating paths learned through iBGP would introduce loops.

A router can either be a source or a sink of packets towards a given destination relative to other routers within the local AS. If the router uses one or more iBGP-learned paths to reach a destination, it's a source and it cannot disseminate any paths of its own over iBGP or packets will loop. Only when all paths in *P* are eBGP-learned paths, a router can be a sink for that destination in the local AS and advertise a path over iBGP. This limitation is expressed in equation 18.

Given this limitation, it would be sub-optimal to accept low-quality iBGP-learned paths in *P*, as these make the router's eBGP paths unavailable for use by the rest of the AS. For this reason, we do not accept iBGP paths in *P* that have an equal or longer AS_PATH than the shortest AS_PATH among eBGP paths in *P*:

$$p \in P^i \wedge q \in P^e \wedge a_p \geq a_q \Rightarrow \{ P \} \setminus p \qquad \forall\, p \in P, \forall\, q \in P^e, p \neq q \quad (9)$$

IV. Loop-freeness

Under normal circumstances, the BGP AS_PATH attribute guarantees loop-freeness. Since our changes allow BGP to use multiple paths concurrently, but only a single path is disseminated to neighboring ASes, checking the AS_PATH for the occurrence of the local AS number is no longer sufficient to avoid loops. Instead, we depend on the Vutukury-Garcia-Luna-Aceves Loop-free Invariant (LFI) conditions [6].

Intuitively, these conditions are very simple: because a router can only use paths that have a lower cost than the path that it disseminates to its neighbors (or, may only disseminate a path that has a higher cost than the paths that it uses), loops are impossible. A loop occurs when a router uses a path that it disseminated earlier, in which case the path that it uses must both have a higher and a lower cost than the path that it disseminates, situations that can obviously not exist at the same time.

When the following two LFI conditions as formulated by Vutukury and Garcia-Luna-Aceves are satisfied, paths are loop-free:

$$FD^i_j \leq D^k_{ji} \quad k \in N^i \qquad (10)$$

$$S^i_j = \{ k \mid D^i_{jk} < FD^i_j \wedge k \in N^i \} \quad (11)$$

"where D^i_{jk} is the value of D^k_j reported to i by its neighbor k; and FD^i_j is the feasible distance *of router i for destination j and is an estimate of D^i_j, in the sense that FD^i_j equals D^i_j in steady state but is allowed to differ from it temporarily during periods of network transitions."* [6]. D^k_j is the distance or cost from router *k* to destination *j*. N^i is the set of neighbors for router *i* and S^i_j is the *successor set* that router *i* uses as next hop routers for destination *j*.

Our interpretation of the two LFI conditions as they relate to BGP is as follows:

$$cp(p_r) < cp_r(p_r) \quad (12)$$

$$P = \{ p \mid cp(p) \leq cp(p_r) \wedge p \in \pi \} \quad (13)$$

Where $cp(x)$ is taken to mean a_x in the case of eBGP and the interior cost for iBGP. The interior cost is the cost to reach a destination as reported by the interior routing protocol that is in use. Because the local AS is added to the AS_PATH as paths are disseminated to neighboring ASes, we swap the smaller and strictly smaller requirements between the two conditions. Figure 3 shows the relationship between the cost (in this case, the AS_PATH length), p_r and equations 12 and 13.

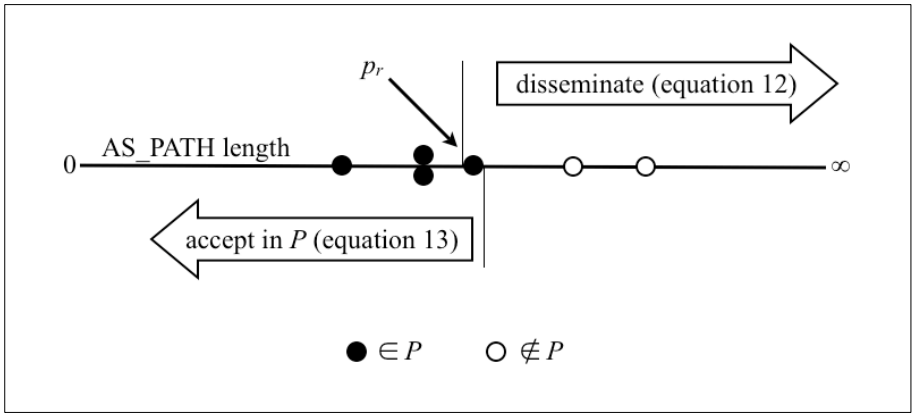

Fig. 3. The relationship between the cost (in this case, the AS_PATH length), p_r and equations 12 and 13.

Equations 10 to 13 are not part of our modified BGP processing rules, as equations 12 and 13 are reformulations of 10 and 11, and equation 14 satisfies both LFI conditions for iBGP while equation 15 satisfies the LFI conditions for eBGP.

A. Loop-freeness for iBGP

Paths learned through iBGP may not be used if the interior cost towards the NEXT_HOP of the path is equal to or larger than the lowest interior cost towards a NEXT_HOP for paths from the multipath set as reported to other routers by the local router:

$$c(h_p) \geq c_r(h_q) \Rightarrow \{ P \} \setminus p \qquad \forall\, p \in P^i, \forall\, q \in P \quad (14)$$

This rule satisfies the requirement imposed by equation 13 for iBGP.

B. AS_SETs

The BGP-4 specification [4] allows for the aggregation of multiple classful destination prefixes into a single CIDR [7] prefix. In that case, the AS numbers in the AS_PATH are replaced with one or more AS_SETs, which contain the AS numbers in the original paths.

```
/* equation 8 */

for each p ∈ P
  for each q ∈ P where p ≠ q
    if (len(AS_PATH(p)) > len(AS_PATH(q)))
      remove p from P

/* equation 9 */

for each p ∈ P
  for each q ∈ P where p ≠ q
    if (p was learned through iBGP and
        q was learned through eBGP and
          len(AS_PATH(p)) ≥ len(AS_PATH(q)))
            remove p from P

/* equation 14 */

for each p ∈ P
  for each q ∈ P where p ≠ q
    if (p was learned through iBGP and
        interior_cost(p) ≥ interior_cost(q))
          remove p from P

/* determine path with longest AS_PATH in P
   for equations 15 and 16 */

longest = NULL

for each p ∈ P
  if (longest ≠ NULL and
      len(AS_PATH(p)) < len(AS_PATH(longest)))
        longest = p
```

```
/* equation 15 /*

if (AS_PATH(longest) contains an AS_SET)
  for each p ∈ P
    if (p ≠ longest)
      remove p from P

/* equation 16 /*

if (! AS_PATH(longest) contains an AS_SET)
  for each p ∈ P
    if (AS_PATH(p) contains an AS_SET)
      remove p from P

for each p ∈ P
  install p in the routing table

/* equation 17, again determine path with
   longest AS_PATH in P */

longest = NULL

for each p ∈ P
  if (longest ≠ NULL and
    len(AS_PATH(p)) < len(AS_PATH(longest)))
      longest_path = p

/* equation 18 */

for each r ∈ neighbouring_routers
  for each q ∈ all_paths_from_all_neighbours
    if (q was learned from r)
      if (q ∉ P)
        disseminate longest to r
      else if (longest was disseminated to r)
        withdraw longest towards r
```

Path selection and dissemination rules for multipath BGP in pseudo-code.

Should the situation arise where a topology is not valley-free and there is both a router that implements multipath BGP as described in this paper as well as, in a different AS, a router that performs aggregation through the use of AS_SETs, then routing loops may be possible. This is so because, depending on the implementation, a router creating an AS_SET could shorten the AS_PATH length and break the limitations imposed by equations 12 and 13.

To avoid these loops, *P* may either contain a single path with an AS_PATH that contains an AS_SET, or no paths with AS_PATHs that contain AS_SETs:

$$a_p = \max(a_p) \wedge set(p) \Rightarrow P = \{ p \} \qquad \forall\, p \in P \quad (15)$$

$$a_p \neq \max(a_p) \wedge set(p) \Rightarrow \{ P \} \setminus p \qquad \forall\, p \in P \quad (16)$$

Note that AS_SETs are rarely used today; a quick count through the Route Views project [9] data reveals that less than 0.02% of all paths have one or more AS_SETs in their AS_PATH.

C. Disseminating Loop-free Paths in eBGP

All paths that remain in the multipath set after the previous steps and after applying policy are installed in the routing table and used for forwarding packets. The determination of traffic split ratios between the available paths is a topic for future work.

At this point, the path with the longest AS_PATH within the set is selected for dissemination to BGP neighbors:

$$a_p = \max(a_p) \Rightarrow p_r = p \qquad \forall\, p \in P \quad (17)$$

Equation 17 satisfies the requirement imposed by equation 12 for both iBGP and eBGP as well as the requirement imposed by equation 13 for eBGP. iBGP uses the interior cost, not the AS_PATH length, as its cost, so equation 17 does not address iBGP.

Through equation 17, multipath-aware ASes will suppress looped paths with a multipath-aware AS in the looped part of the path, while regular BGP AS_PATH processing suppresses looped paths with no multipath-aware ASes in the looped part of the path.

If multiple paths share the maximum AS_PATH length, the path that was previously disseminated to BGP neighbors, if any, is selected for dissemination. This has the effect of damping oscillations on shorter paths.

D. Loop-freeness for Multipath-unaware iBGP Routers

To avoid loops for non-multipath-aware iBGP routers, the selected path is also not disseminated over any BGP session over which the router learned a path that is in the multipath set:

$$q \notin P \Rightarrow \text{disseminate } p_r$$

$$q \in P \Rightarrow \text{withdraw / do not disseminate } p_r$$

$$\forall\, r \in R,\ \forall\, q \in \pi,\ r = r_q \quad (18)$$

If the router previously disseminated a path over a session towards a neighboring router that supplied a path in the selected multipath set *P*, it now sends a withdrawal for the multipath destination.

The pseudo code lists the complete set of rules that accomplish the multipath processing required to install multiple paths in the routing and forwarding tables, and to prevent loops.

V. Convergence

Intuitively, it's easy to see that BGP topologies with conflicting policies have trouble converging [10]. For instance, if A prefers to send traffic through B, while B prefers to send traffic through A, BGP's loop detection will make sure that both do not happen at the same time, but it will not be possible to reach a stable, converged state: the final state depends on the order of events.

On the other hand, when the Gao-Rexford guidelines are observed, convergence to a stable state is guaranteed because in that case, there are no cycles in the configured policies. This means that whenever an AS selects a path, decisions made subsequently by upstream ASes will not make the earlier AS select a different path.

Like standard BGP, our path selection rules require that only paths with the highest LOCAL_PREF are included in the candidate route set (*P*). Because each LOCAL_PREF value maps to a single valley-free class (sibling, service provider, peer or customer), our use of multiple paths does not break the valley-free property if it was present in the single path case.

So, in the case of valley-free topologies, eventual convergence is guaranteed and largely the same as that for a topology where only the longest paths used exist. However, there may be more intermediate states and updates for those states may trigger the minimum route advertisement interval, pushing convergence times towards the maximum imposed by this interval. Non-valley-free topologies may never converge. The presence of longer paths

injected by multipath-aware routers may exacerbate this situation as the multipath-aware routers try to find the longest loop-free paths allowed by policy. The quantification of these effects is part of our future work agenda.

VI. Evaluation

In order to evaluate the impact of our changes to BGP, in particular to get a grasp on the dynamics of the resulting system, we created a simulator that implements our modified BGP rules [22]. The simulator is a script that outputs the result of (multipath) BGP decision making based on a given input topology. The script can also simulate the decision making in traditional BGP routers. Figure 4 shows an example topology with all the routers using the existing BGP path selection and tie breaking rules. Each circle is an AS with a single router in it. Only the router in AS 7 announces a prefix. The solid arrows indicate the path selected by each AS, with the dotted arrows indicating additional paths present in the BGP table but not used. Note that in each case, unused paths exist in both directions.

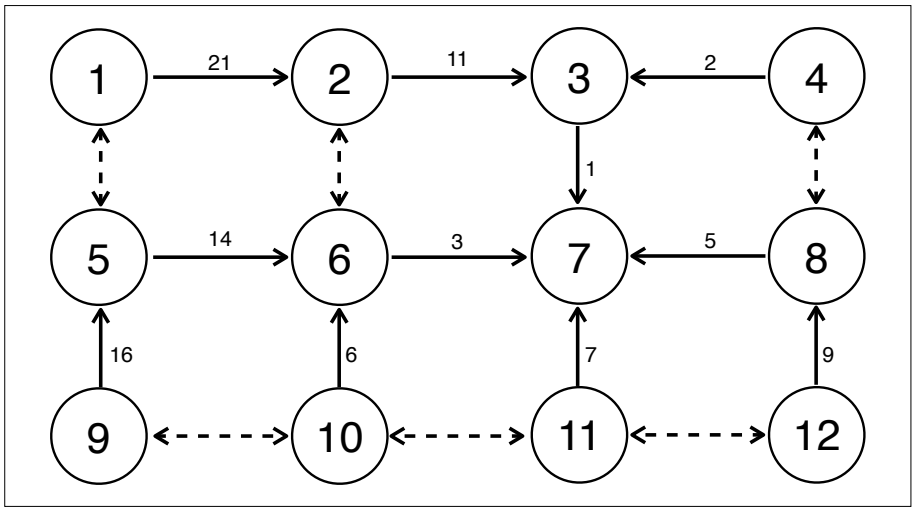

Fig. 4. Preferred unmodified BGP reachability and backup paths to AS 7.

The numbers adjacent to the arrows are the iteration numbers for the iteration when this path was selected. Note that the definition of iteration for this purpose is such that only a single router makes a path selection decision about a single path, in reality many decisions are made in parallel due to the distributed nature of the Bellman-Ford algorithm that underlies all distance vector protocols and thus BGP. Figure 5 shows the same topology as figure 4, but now all routers are multipath enabled and β is set to infinity. Each arrow indicates a path used for forwarding packets, the heavy arrow indicates the path disseminated to neighbors.

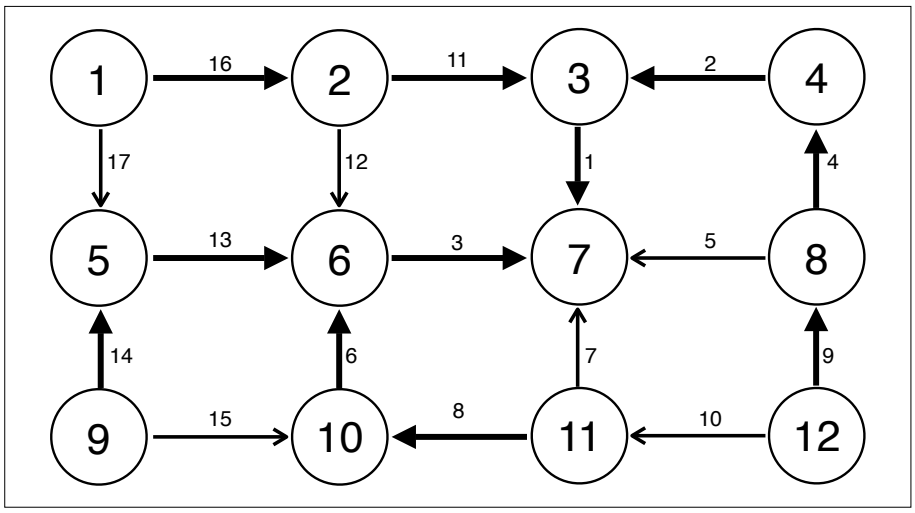

Fig. 5. Preferred multipath reachability to AS 7, β = ∞

Interestingly, the number of iterations needed for multipath BGP to converge is actually slightly lower than the number of iterations needed by traditional BGP. This is probably because each router greedily obtains all the paths that it can, limiting the choices of other autonomous systems.

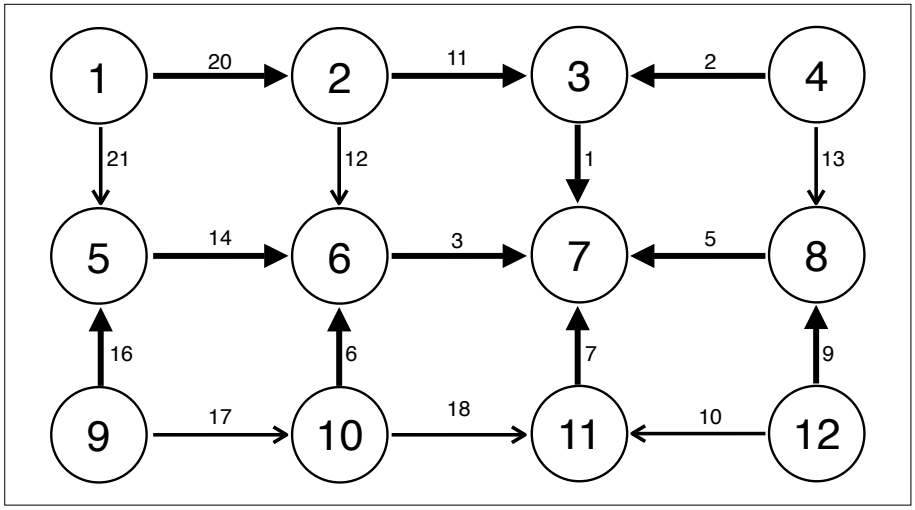

Fig. 6. Preferred multipath reachability to AS 7, β = 1

Figure 6 shows the same topology with multipath BGP enabled, but now with a value of 1 for β, so a router will not select any paths that have an AS_PATH that is more than one AS hop longer than the shortest available AS_PATH. In this case, only paths with equal length AS_PATHs are selected. In this case, AS 12 selects path 8-7 as the path that it disseminates to neighbors (the path with the longest AS_PATH) while in figure 5, this was the path 8-4-3-7. So excessively long paths are avoided, while half of the ASes are still capable of using a second path. However, the number of iterations required to converge is back to 21, the same as traditional BGP.

While much further understanding of the dynamics is needed, the obtained results are promising, since the results from the experiments performed show that the proposed multipath BGP converges in a similar (if not smaller) number of iterations as current BGP and that it manages to avoid long paths, all this in challenging topologies.

VII. Related work

The most common multipath mechanisms are the ones existing for intra-domain protocols, like the ECMP (Equal Cost Multipath) in OSPF [2] or IS-IS, the unequal cost multipath in IGRP and EIGRP [3] or provided by means of basic source routing mechanisms. All these proposals to select multiple paths inside an AS may work together with the solution proposed in this paper, which is mainly issued for inter-domain routing.

Some multipath proposals for inter-domain routing are following the intra-domain alternatives like the source routing proposals in [12] or [13] or the proposal based on overlays like MIRO [14]. As discussed in [14], the source routing proposals are in general too restrictive for the intermediate ASes whose flexibility to decide on alternative routes is reduced. Regarding the overlay solutions, they normally imply an additional complexity associated with the tunneling mechanisms and the overhead that the tunnels introduce.

MIRO however, reduces the overhead during the path selection phase by means of a co-operative path selection involving the different intermediate ASes (additional paths are selected on demand rather than disseminating them all every time). The proposal made in this paper is not requiring an overlay for the multipath mechanism to work, making deployment easier, since it does not requires changes in the neighboring ASes.

Finally, it is important to reference some proprietary solutions that are already providing multipath BGP for commercial equipment like Cisco [15] or Juniper [16]. As the solution proposed in this article, Cisco's solution respects BGP semantics.. It is, however, too restrictive with the conditions that a path must fulfill in order to be selected (extra paths are almost equal to the best one). Juniper's solution is oriented towards the provisioning of backup links and load balancing between adjacent BGP peers than towards the provisioning of a disjoint set of paths towards a certain destination.

VIII. Conclusion and future work

Multipath inter-domain routing is a powerful tool that results in substantial advantages, including increased network capacity, enhanced redundancy and better response to congestion events. We have shown that, contrary to the limitations accepted in common practice, it is possible to accept multiple paths for forwarding packets without risk of routing loops. This can be achieved without changes in the BGP semantics and only requiring local changes in the BGP route processing mechanism. This results in a powerful deployment model based in the incentive vector where the party that deploys the mechanism is the party that gets the benefits.

However, additional research is needed to fully understand the impact of the proposed mechanism. First, the resulting dynamics of the proposed BGP multipath approach need further investigation. In particular, even if we know that the proposed modification does not change the convergence result (i.e. configurations that converge in regular BGP still converge in our proposed multipath approach), additional analysis is required in how the proposed changes affect the convergence process, including a quantification of the expected number of iterations to converge. In addition, we need to quantify the increase in the stability of the resulting paths. As we mentioned before, shorter path changes are no longer propagated, so there is potential reduction in routing churn that needs to be quantified.

Another aspect that needs more research is the resulting path distribution and diversity if the proposed mechanism is widely implemented. Other approaches to multipath BGP would

be to disseminate AS_SETs containing all the ASes in AS_PATHs of all paths used, and changing BGP such that multiple paths can be communicated between two neighbors.

Acknowledgments

This thesis is based on a yet unpublished paper written with Jon Crowcroft, Marcelo Bagnulo and Francisco Valera.

The research results presented herein have received support from Trilogy (http://www.trilogy-project.org), a research project (ICT-216372) partially funded by the European Community under its Seventh Framework Program. The views expressed here are those of the author(s) only. The European Commission is not liable for any use that may be made of the information in this document.

References

[1] F. Kelly, T. Voice. Stability of end-to-end algorithms for joint routing and rate control. Computer Communication Review 35:2, 2005.

[2] J. Moy. OSPF Version 2. RFC 2328, April 1998.

[3] R. Albrightson, J.J. Garcia-Luna-Aceves, and J. Boyle. EIGRP—A Fast Routing Protocol Based On Distance Vectors. Proc. Networld/Interop 94, May 1994.

[4] Y. Rekhter, T. Li, S. Hares. A Border Gateway Protocol 4 (BGP-4). RFC 4271, January 2006.

[5] M. Mitzenmacher. The Power of Two Choices in Randomized Load Balancing. PhD Thesis, Harvard, 1996.

[6] S. Vutukury, J.J. Garcia-Luna-Aceves. A Simple Approximation to Minimum-Delay Routing. Proc. of ACM SIGCOMM, 1999.

[7] V. Fuller, T. Li. Classless Inter-domain Routing (CIDR): The Internet Address Assignment and Aggregation Plan. RFC 4632, August 2006.

[8] L. Gao, J. Rexford. Stable Internet Routing Without Global Coordination. Proc. ACM SIGMETRICS, June 2000.

[9] Route Views project. <http://www.routeviews.org/>, July 2008.

[10] K. Varadhan, R. Govindan, D. Estrin. Persistent route oscillations in inter-domain routing. ISI technical report 96-631, USC/Information Sciences Institute, 1996.

[11] C. Hopps. Analysis of an Equal-Cost Multi-Path Algorithm. RFC 2992, November 2000.

[12] D. Zhu, M. Gritter, D. Cheriton. Feedback based routing. Proc. Hotnets, 2002.

[13] H. T. Kaur, S. Kalyanaraman, A. Weiss, S. Kanwar, A. Gandhi. BANANAS: An evolutionary framework for explicit and multipath routing in the Internet. Proc. Future Directions in Network Architecture, 2003.

[14] W. Xu, J Rexford. MIRO: multi-path interdomain routing. Proc. ACM SIGCOMM 2006.

[15] BGP best path selection algorithm. Document ID: 13753. May 2006. Cisco Systems. <http://www.cisco.com/application/pdf/paws/13753/25.pdf>. Last visited jul-2008.

[16] J. M. Soricelli. Juniper™ Networks Certified Internet Specialist Study Guide 2004. Juniper Networks. ISBN: 0-7821-4072-6

[17] I. van Beijnum. BGP: Building Reliable Networks with the Border Gateway Protocol. O'Reilly & Associates, Inc., Sebastopol, CA, 2002. ISBN: 0-596-00254-8.

[18] L. Gao. On inferring autonomous system relationships in the Internet. IEEE/ACM Trans. on Networking, 2000.

[19] S. Y. Qiu, P. D. McDaniel, F. Monrose. Toward valley-free inter-domain routing. IEEE ICC, 2007.

[20] R. Chandrasekeran, P. Traina, T. Li. BGP Communities Attribute. RFC 1997, August 1996.
[21] Internet Assigned Numbers Authority. Border Gateway Protocol (BGP) Well-known Communities. <http://www.iana.org/assignments/bgp-well-known-communities/>. April 2004.
[22] I. van Beijnum. runbgp script and example topologies. <http://www.bgpexpert.com/runbgp.tar.gz>. September 2008.